Collage of Computers and Information Technology
Project ID: <501598-3.>
Date: <08/09/2022>

Digital Twin-Assisted Physiotherapy for Spine Disc Rehabilitation

Atouf Saleh Alazwari (44103542)

Shatha Abdullah Alshamrani (44102808)

Hadeel Mohammed Altalhi (44103008)

Shahad Massoud Althobaiti (44107446)

Ruba khalid Aloufi (44104338)

Shaikhah Obaid Ba Lhamar (44151334)

Retaj Adnan Altwirqi (44100431)

Fatimah Safar Alotaibi (44103854)

Supervised by:

Dr. Wael Y. Alghamdi

Department of Computer Engineering

Collage Computers and Information Technology

Taif University, KSA

October 2024

Contact Information

Authors Information		
Name	Number	Email
1. Atouf Saleh Alazwari	44103542	s44103542@students.tu.edu.sa
2. Shatha Abdullah Alshamrani	44102808	s44102808@students.tu.edu.sa
3. Hadeel Mohammed Altalhi	44103008	s44103008@students.tu.edu.sa
4. Shahad Massoud Althobaiti	44107446	s44107446@students.tu.edu.sa
5. Ruba khalid Aloufi	44104338	s44104338@students.tu.edu.sa
6. Shaikhah Obaid Ba Lhamar	44151334	s44151334@students.tu.edu.sa
7. Retaj Adnan Altwirqi	44100431	s44100431@students.tu.edu.sa
8. Fatimah Safar Alotaibi	44103854	s44103854@students.tu.edu.sa

Supervisor

Name: Dr. Wael Y. Alghamdi

Department of Computer Science

Students' Property Right Declaration

I hereby declare that the work in this capstone project at Taif University is my own except for quotations and summaries which have been duly acknowledged. This work with the title **Digital Twin-Assisted Physiotherapy for Spine Disc Rehabilitation** has not been accepted for any degree and is not concurrently submitted for award of other degrees. It is the sole property of Taif University and it is protected under the intellectual property right laws and conventions.

Authors:

Name	Signature:	Date:
Atouf Saleh Alazwari		10/04/2024
Fatimah Safar Alotaibi		10/04/2024
Hadeel Mohammed Altalhi		10/04/2024
Retaj Adnan Altwirqi		10/04/2024
Ruba khalid Aloufi		10/04/2024
Shahad Massoud Althobaiti		10/04/2024
Shaikhah Obaid Ba Lhamar		10/04/2024
Shatha Abdullah Alshamrani		10/04/2024
Supervisor:		
Dr. Wael Y. Alghamdi	———	10/04/2024

Students Anti-Plagiarism Statement

I hereby declare this report is my own work except for properly referenced quotations, and contains no plagiarism; it has not been submitted previously for any other assessed unit on this or other degree courses.
I have read and understood the School's rules on assessment offences which are available in the Taif University Handbook

انا الممضي أسفله أشهد أن هذا التقرير هو عملي الخاص انا و مجموعة الطلبة المذكورة أسماؤهم بأول هذا التقرير ما عدا ما هو مذكورة مصادره صراحةً و أنه لا يحتوي على محتويات منقولة بدون عزوها لكاتبها الأصليّ. و أشهد أنّ هذا العمل لم يسبق أن استخدم كعمل رسمي بمقررات اخرى بهذه الكلّية أو غيرها.

أشهد أنّي اطّلعت على قوانين الكلية الخاصّة بتقييم الطّلبة الموجود بالكتاب التقديمي للجامعة.

Authors:

Name	Signature:	Date:
Atouf Saleh Alazwari		10/04/2024
Fatimah Safar Alotaibi		10/04/2024
Hadeel Mohammed Altalhi		10/04/2024
Retaj Adnan Altwirqi		10/04/2024
Ruba khalid Aloufi		10/04/2024
Shahad Massoud Althobaiti		10/04/2024
Shaikhah Obaid Ba Lhamar		10/04/2024
Shatha Abdullah Alshamrani		10/04/2024

Abstract

A digital twin is a dynamic virtual model of a physical object, system, or process that continuously mirrors its real-world counterpart through data exchange. It relies on live data, simulations, and artificial intelligence to monitor, optimize, and support decision-making across various fields. In healthcare, digital twins are revolutionizing patient care by integrating data from multiple sources to enable personalized treatments and predictive analyses. They are used to create virtual models of organs that assist in surgical planning and reduce potential harm. Moreover, digital twins simulate patient health pathways to enhance diagnosis and treatment outcomes. A prominent example is the development of a digital twin for the heart, which facilitates customized surgeries and improves diagnostic accuracy. In clinical research, digital twins simulate patient outcomes, aiding in both improved care and drug development. Digital twin technology also offers significant potential in simulating the biomechanics of the spine, enabling predictions of how the spine will respond to various treatments. By integrating computational models with real-time data, digital twins can enhance personalized treatment planning.Hence, we proposed the idea of using a digital twin to help the physical therapist simulate exercises on the digital twin before applying them to the patient, allowing the therapist to evaluate the effects of each movement and estimate the period of the treatment plan. Using a smart vest supported by ECG sensor and flex sensor that measures curvatures and analyzes pain levels. It will be the point of contact for transferring data between the physical twin and the digital twin.

Keywords— Digital twins , ECG sensor ,flex sensor ,simulations biomechanics ,virtual models , real-time data.

المســـتخـــلص

التوأم الرقمي هو نموذج افتراضي ديناميكي لجسم أو نظام أو عملية فيزيائية، يعكس بشكل مستمر نظيره في العالم الواقعي من خلال تبادل البيانات. ويعتمد على البيانات الحية، المحاكاة، والذكاء الاصطناعي لمراقبة وتحسين واتخاذ القرارات عبر مختلف المجالات. التوأم الرقمي يُحدث ثورة في الرعاية الصحية من خلال دمج بيانات المرضى من مصادر متعددة لتقديم العلاج الشخصي والتحليلات التنبؤية. يُستخدم في إنشاء نماذج افتراضية لأعضاء الجسم تُساعد في التخطيط للجراحات وتقليل الضرر، كما أنه يُحاكي مسارات صحة المرضى لتحسين التشخيص والعلاج. من أبرز الأمثلة هو تطوير توأم رقمي للقلب لتخصيص العمليات الجراحية وتحسين دقة التشخيص. كما يساهم في الأبحاث السريرية من خلال محاكاة نتائج المرضى، مما يساعد في تحسين الرعاية وتطوير الأدوية. تقدم تقنية التوأم الرقمي إمكانيات كبيرة في محاكاة الميكانيكا الحيوية للعمود الفقري، مما يسمح بالتنبؤ باستجابة العمود للعلاجات المختلفة. من خلال دمج النماذج الحسابية مع البيانات في الوقت الفعلي، يمكن للتوائم الرقمية تحسين التخطيط العلاجي الشخصيومن هنا اقترحنا فكرة استخدام توأم رقمي يساعد المعالج الفيزيائي في محاكاة التمارين على التوأم الرقمي قبل تطبيقها على المريض ، مما يسمح للمعالج بتقييم الآثار المترتبة على كل حركة . باستخدام سترة ذكية مدعمه بحساسات لقياس ضربات القلب بالاضافة الى حساسات المرونة التي تقوم بقياس الإنحناءات وتحليل مستوى الألم . حيث أنها ستكون نقطة الوصل لنقل البيانات بين التوأم الفيزيائي والتوأم الرقمي .

الكلمات المفتاحية التوائم الرقمية، مستشعر تخطيط القلب الكهربائي، مستشعر الانحناء، المحاكاة، الميكانيكا الحيوية، النماذج الافتراضية، البيانات في الوقت الفعلي.

Table of Contents

List of Tables

List of Figures

A digital twin is a dynamic virtual model of a physical object, system, or process that continuously mirrors its real-world counterpart through data exchange. It relies on live data, simulations, and artificial intelligence to monitor, optimize, and support decision-making across various fields. In healthcare, digital twins are revolutionizing patient care by integrating data from multiple sources to enable personalized treatments and predictive analyses. They are used to create virtual models of organs that assist in surgical planning and reduce potential harm. Moreover, digital twins simulate patient health pathways to enhance diagnosis and treatment outcomes. A prominent example is the development of a digital twin for the heart, which facilitates customized surgeries and improves diagnostic accuracy. In clinical research, digital twins simulate patient outcomes, aiding in both improved care and drug development. Digital twin technology also offers significant potential in simulating the biomechanics of the spine, enabling predictions of how the spine will respond to various treatments. By integrating computational models with real-time data, digital twins can enhance personalized treatment planning.Hence, we proposed the idea of using a digital twin to help the physical therapist simulate exercises on the digital twin before applying them to the patient, allowing the therapist to evaluate the effects of each movement and estimate the period of the treatment plan. Using a smart vest supported by ECG sensor and flex sensor that measures curvatures and analyzes pain levels. It will be the point of contact for transferring data between the physical twin and the digital twin.

Chapter 1 | Introduction

1.1 Overview and Problem Background

A Digital Twin is a dynamic virtual representation of a physical object, system, or process that continuously mirrors its real-world counterpart through data exchange. First introduced by Michael Grieves in 2002 [1], Digital Twins are virtual models of physical systems or objects. These models, driven by real-time data, simulations, and machine learning, enable monitoring, optimization, and decision-making across various industries.

In manufacturing, Digital Twins simulate production lines to boost efficiency and minimize downtime. Similarly, healthcare employs Digital Twins to personalize treatment by simulating organ models before surgeries. Smart cities utilize these virtual replicas to manage infrastructure, traffic, and energy resources effectively. In cybersecurity, Digital Twins can forecast incidents through AI analysis, offering a virtual space for detecting system failures and cyberattacks. However, their enhanced connectivity introduces vulnerabilities, particularly in construction, where they expose infrastructure to potential cyber threats. Moreover, they are crucial for simulating cyber-physical systems to enhance responses to security threats.

Industries such as automotive, aviation, and healthcare have significantly benefited from Digital Twins. For example, BMW uses them to optimize global factory operations and supply chains, while in aviation, they aid in production and maintenance. Healthcare applications of Digital Twins include studying brain atrophy and simulating heart functions. Regular updates to these models are facilitated by technologies like big data and artificial intelligence, sourced from sensors and wearable devices. Digital Twins are transforming industries by bridging the gap between the physical and digital worlds, offering real-time optimization, decision-making, and testing that drive innovation and efficiency across sectors.

1.2 Problem Statement

Herniated disc is a significant condition affecting adults, particularly those aged 31 to 60 years. Despite representing less than 5% of low back pain (LBP) causes, it leads to severe discomfort and disability. The global prevalence is 5 to 20 cases per 1000 adults annually, with LBP being a leading cause of disability. Certain lifestyle factors, like prolonged sitting, inactivity, and heavy lifting, contribute to the rising number of herniated disc cases [2].

The Role of Physical Therapy in Recovery Physical therapy plays a critical role in managing herniated disc-related LBP. It includes strengthening exercises, coordination, and endurance training, which significantly reduce pain and improve function. Tailored therapy based on symptom patterns provides better outcomes, particularly in mechanical low back pain, where 90% of patients return to work after treatment. However, patients with herniated discs may need longer treatment periods to see improvement, although early intervention generally leads to better results [3].

1.3 Objectives

Using digital twins, we provide online and offline simulation to the human body to allow physio therapist to make the right moves during the session and estimate the period of recovery. In offline mode, we will simulate the patient's movements to analyze their effect on the disc and nerves. If a movement proves suitable and beneficial for the patient's condition, we will proceed with it in online mode to observe real-time results and monitor the patient's progress.

Thus, this research objective's are as follows:

- Provide a product to physiotherapist to try different moves to the digital twin's patient before applying them into the actual patient.

- Estimate the actual recovery period.
- Avoid wrong movement which may cause negative affect on patients.

In summary, by using digital twins technology, we can identify which movements the patient can perform without experiencing pain, and which movements will improve their condition and speed up recovery. This approach will ensure the patient's condition improves without the risk of further injury from incorrect movements, which could potentially harm the discs or nerves.

1.4 Contributions/Significance of the Project

Digital Twins revolutionize industrial operations by creating a real-time bridge between physical systems and digital environments. They allow for continuous optimization and testing, which reduces costs and enhances decision-making. By simulating digital twins for spine/disk patients, we look for enhancing treatment as well as estimating the recovery period. Importantly, we seek avoiding wrong movements which may has a bad affects on patients.

1.5 Engineering Standards

Below an example of ”Engineering standards”:
-The International System of Units (SI):
-Time (Seconds),
-Voltage (Volt),
-Current (Ampere),
-Electric Resistance (Ohm Î©),
-Frequency (Hz).

-Temperature (Degree Celsius CÂ°),
-Angle (Degree Â°) - although not an SI unit it is mentioned as an accepted unit
-Torque (Newton-Meter N-m).
-RS-232: we used this serial communication standard to connect the Arduino board with PC in order to upload the program.
-Bluetooth 4.0: We used this standard for wireless communication between two devices in our project.
-NEMA 17 Stepper Motor: 4-wire, bipolar, step angle1.8o, current 1.7A.
-Red-Yellow-Green: We used this color standard in our traffic lights project.

1.6 Realistic Constrains

-Economic: effect of this topic on economy, possible cost of project development, cost of materials, target cost if project is marketed.
-Environmental: Influence on the environment, possible effects for future developments.
-Sustainability: product life cycle, future markets.
-Manufacturability: materials availability, special needs for hostile environments.
-Ethical: ethical issues that someone working on this topic might encounter.
-Health and safety: positive or negative impacts on the health and safety individuals for past or future applications.
-Social: relationship of this topic to social aspects of society such as education, culture, communication, entertainment.

1.7 Development Time Frame and Cost

The schedule is a significant part of the project. It defines what you intend to do and when you plan to do it. You should consider how long each activity will take, which activities must precede others, and how much overlap is possible or desirable. The schedule identifies tasks to be performed, milestones to be met, and the estimated number of hours for each task.

Table 1.1: Semester 1 Schedule Tasks

Week %	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	Hrs
Tack																
Project Proposal	12	12	8	8	8	8									56	25.9
Study of current research work			4	4	4	4									16	7.4
Problem formulation			4	4	4	4									16	7.4
Model design							16	16	12	12	12				68	31.5
CP1 Report									4	4	4	12	12		36	16.7
Presentation												4	4	16	24	11.1
Total Hours	12	12	16	16	16	16	16	16	16	16	16	16	16	16	216	100

Table 1.2: Semester 2 Schedule Tasks

Week %	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	Hrs
Tack																
System Implementation	16	16	16	16	8	8	8	4						92	43.4	
System integration and Testing					8	8	8	8							32	15.1
Final Report									4	16	16	8	8		52	24.5
Project Poster												4	4	4	12	5.7
Presentation 11.3												4	4	8	8	24
Total Hours	16	16	16	16	16	16	16	16	16	16	16	16	12	8	212	100

1.8 Conclusion

In conclusion, using digital twin technology to support physiotherapists in treating disk or spine problems is essential for improving therapeutic outcomes. digital twin allows for the identification of safe movements that contribute to faster recovery and prevent injuries caused by incorrect movements. Additionally, it enables precise monitoring of the patient's condition and real-time adjustments to ensure effective care.

This places your key sentence at the beginning as requested. This chapter highlights:

- The foundational concept of Digital Twins as a dynamic virtual model of physical systems.
- Their role in various industries, including manufacturing, healthcare, and smart cities.
- The contribution of Digital Twins to enhancing efficiency, decision-making, and security.
- The potential cybersecurity risks introduced by their increased connectivity.
- The need for further research and development to optimize the integration of AI, big data, and simulation technologies.

Finally, Appendix in Chapter 6 shows the demo explaining the idea as well as the prototype.

Chapter 2 | Literature Review

2.1 Digital Twin Overview

A Digital Twin is a dynamic virtual representation of a physical object, system, or process, created to mirror its real-world counterpart through continuous data exchange. Introduced by Michael Grieves in 2002 [1], the Digital Twin is a dynamic virtual model of physical objects or systems. Through real-time data, simulations, and machine learning, Digital Twins enable monitoring, optimization, and decision-making across various industries. In manufacturing, Digital Twins simulate production lines, enhancing efficiency and reducing downtime [4]. In healthcare, they allow for personalized treatment by simulating organ models before surgery [5]. Smart cities utilize Digital Twins to optimize resource management, modeling infrastructure, traffic, and energy usage [6].

In cybersecurity, Digital Twins help predict incidents by analyzing data with AI, providing a virtual space to detect system failures and cyberattacks [7]. However, their increased connectivity introduces risks, especially in construction, where digital models expose infrastructure to cyber threats [8]. They also simulate cyber-physical systems to improve response to security threats [9]. Industries like automotive, aviation, and healthcare benefit from Digital Twins. BMW uses them to optimize global factory operations and supply chains [10], while aviation applies them to improve production and maintenance [11]. In healthcare, Digital Twins are used to study brain atrophy and simulate heart functions [12] [13].

Digital Twins are transforming industries by bridging the physical and digital worlds. They enable real-time optimization, decision-making, and testing, driving efficiency and innovation across sectors.

However, despite their broad potential, limited work has been done on Digital Twins across different industries. In the manufacturing sector, Bottani et al. (2017) presented a prototype Digital Twin of automated guided vehicles to optimize the adaptive behavior of production systems. Brenner and Hummel (2017) explored how Digital Twins can be harnessed for shop floor management systems in a logistic learning factory. Schleich et al. (2017) introduced a conceptual framework of a Digital Twin model that integrates the design and manufacturing processes in production engineering, further highlighting the growing role of Digital Twins in bridging data transmission and coordination between physical artifacts and digital models [14].

Figure 2.1: Overview of Innovative Projects in Digital Twins Technologies [15]

2.2 Using Digital Twin in Medical Field

Digital twins are revolutionizing healthcare by integrating patient data from various sources, including wearable medical video recordings. This technology enables personalized treatment and predictive analytics of disease progression and complications, leading to significant healthcare innovations [16].

Digital twins are virtual models representing internal bodies, used in holistic medicine, epidemiological data scanning, and real-time customer data. With the lack of choice of internet devices, continuous selection of physiological and environmental data. These data are processed using medical records and electronic health forms and help facilitate university study. In addition to a single-digit digital multi-digit result system, which supports the possibility of digital healthcare [17].

In surgery, digital twin technology features virtual models of patients that help plan surgical procedures and minimize damage to vital organs. Specialized medical teams can customize surgical procedures and confirm anatomical details prior to specialized surgeries [18].

One of the best efforts is the development of a digital twin of the heart using rapid effects unit techniques. This activity is active in cardiac modeling, blood technology, and new vision, and facilitates the identification of the need for unique global integrations and accurate outcomes quickly [19].

Digital Twin Approach for Modeling Brain Atrophy in Multiple Sclerosis. This research uses a digital twin to create a model for understanding brain atrophy onset in multiple sclerosis. By comparing thalamic atrophy data with a virtual healthy model, it identifies biological changes before clinical symptoms appear, aiming to improve early diagnosis and personalized treatment [20].

A digital twin in healthcare represents a patient or a medical system that reflects conditions in the world through the exchange of real-world data. This component combines patient data in the practical and virtual text and precision medicine analytics, which can help improve surgical procedures by simulating the responses of different organs [21].

Digital Twins (DTs) in clinical research are virtual models that replicate patients' health trajectories using data from the individual. These models simulate outcomes using advanced algorithms, allowing a bidirectional flow of information between the patient and the DT. The DT provides real-time predictions, such as identifying potential adverse events, and supports clinical decision-making. The goal is for DTs to closely mirror patients' clinical variables and disease progression, enhancing both patient care and drug development [22].

2.3 Using Digital Twin in Spine and Disc

Digital twin technology offers significant potential in simulating the bio-mechanics of the spine and predicting how it might respond to various treatments or surgeries. By integrating advanced computational models with real-time data acquisition, digital twins can provide a dynamic and accurate representation of a patientâs spine, allowing for personalized treatment planning where the specific bio-mechanical properties of an individual's spine are considered. This approach can lead to more precise and effective interventions. For example, one study proposed an integrated digital twin solution for lumbar spine analysis that combines shape and performance data using artificial intelligence (AI), motion capture technology, and finite element methods (FEM) to predict joint contact forces and intradiscal pressure in real time. Participants in this study wore virtual reality equipment and sensors to capture their movements, which were then analyzed to estimate bio-mechanical stresses during various motions, such as flexion and extension. Although this study did not achieve a fully mature digital twin due to limitations in real time data acquisition, it lays the groundwork for future advancements in digital twin technology for spinal health [23].

Several research efforts have explored the application of digital twins in improving the diagnosis and treatment of spinal conditions, such as disc herniation and scoliosis. For instance, an AI-based framework known as ReconGAN has been developed to create digital twins of vertebrae to enhance fracture risk predictions, particularly in patients with spinal cancer. This framework integrates bone microstructure models generated from imaging data with finite element analysis, offering a promising tool for personalized treatment strategies in spinal health [24]. Another example is the development of the Surgical Digital Twin (SDT) system, which aims to create an accurate virtual copy of surgical procedures using modern technologies. This systemâs primary objective is to improve surgical planning and training, providing a proof of concept (PoC) for how digital twins can be utilized in spinal surgery. This approach not only enhances the precision of surgical procedures but also generates valuable training data for future surgeons [25].

In addition to these efforts, the SPINNER project, led by the University of Sheffield, further demonstrates the potential of digital twin technology in spinal health. This program is designed to train bioengineers in the development of new materials and techniques for spinal repair, with a focus on improving surgical outcomes and increasing healthcare efficiency. Through digital twin simulations, participants can replicate patient cases, providing real-time support to surgeons and aiding in more informed medical decision-making. This integration of digital twin technology into surgical training highlights the growing importance of virtual models in enhancing both the precision and success rates of spinal procedures [26].

Digital twin technology can significantly improve treatment outcomes by providing tailored healthcare solutions for each patient. By simulating the functional performance of the spine in a virtual environment, healthcare providers can better understand the impact of various treatments before they are administered. For instance, digital twin models have been used in other fields, such as modeling centrifugal valves in âlab-on-a-discâ systems, to predict performance and system stability. This approach allows for optimization in a virtual setting, reducing the need for expensive and time-consuming physical prototyping [27]. Similarly, in spinal health, digital twins can help refine treatment plans by simulating different scenarios, ultimately reducing the likelihood of surgical interventions by offering less invasive alternatives based on real-time data.

One of the most promising aspects of digital twin technology in spinal care is its ability to reduce the necessity for surgical intervention. By providing a highly accurate simulation of the spineâs biomechanics, digital twins allow for the exploration of non-surgical treatment options that may achieve the desired outcomes. Treatments can be tested and optimized in the virtual environment before being applied to the patient, ensuring that only the most effective and least invasive methods are pursued. This data-driven approach not only improves patient outcomes but also reduces the risks associated with surgery, such as infection and recovery time [28].

2.4 Research Gap

The project would be the first Digital Twin that assists physical therapist to predict the extent of the effect of movements on the spine, especially for patients with herniated discs

Chapter 3 | System Analysis and Design

3.1 Overview and System Model

This section provides a brief overview of the problem context and introduces the proposed system model. It highlights the key challenges addressed and outlines the system's structure, including its components and functionalities. The system model is designed to provide an efficient, scalable, and innovative solution to the identified problem, ensuring accuracy and reliability in its application.

The digital twin project aims to assist physiotherapists through the use of a smart vest, which helps the therapist measure pain intensity. This innovative tool facilitates communication between the physical and digital twins, enabling the system to analyze performance and customize therapeutic exercises to suit the patientâs condition. This approach enhances treatment efficiency while minimizing the risks associated with directly applying exercises.

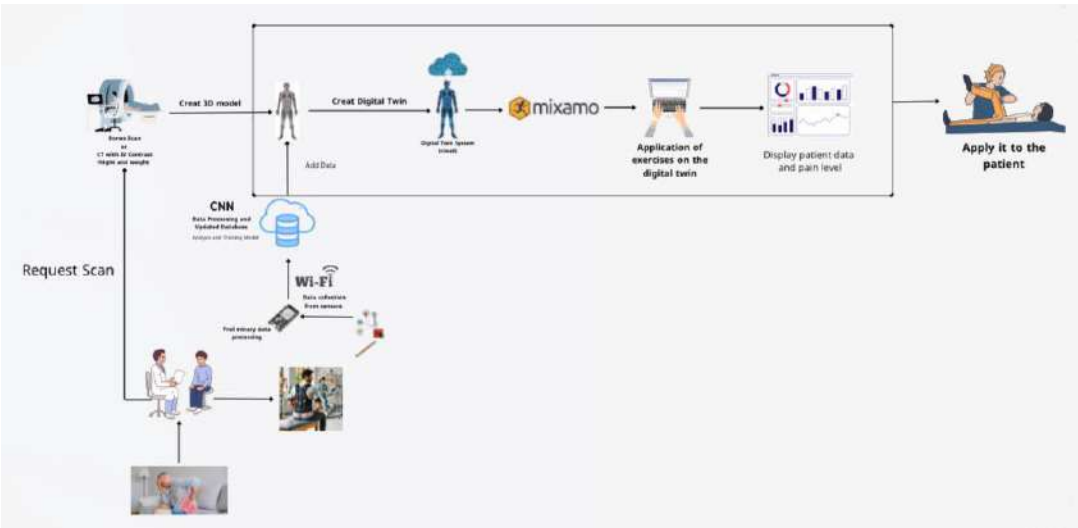


Figure 3.1: System Architecture Model

In Figure 3.1, the system consists of several interconnected components, including: Request Scan,Create 3D Model,Smart Jacket,ESP32Microcontroller, MQTT Protocol, Data Processing and Updated Database, Create Digital Twin, Blender Software, Application of Exercises on the Digital Twin, Clearly Display Results and Analyses to the Physical Therapist, and Apply it to the Patient.

When the patient arrives at the clinic, the physical therapist accesses the system and requests an appropriate scan for the patient based on their condition, such as a Bone Scan or CT with IV Contrast. This scan data is directly used to create a 3D model representing the patientâs current health status accurately. After completing this phase, the physical therapist prepares the patient by equipping them with the Smart Jacket equipped with ECG and Flex Sensors, which immediately start collecting biomechanical and physiological data such as posture deviations and pain levels during movement. The collected data from the sensors is sent to the ESP32 microcontroller, which processes the signals and converts them into usable digital data. This data is securely transmitted via the MQTT protocol to ensure data safety and real-time flow. The encrypted data is then sent to the cloud database for analysis and updating the 3D model with new information, along with the scan data. The updated data is applied to the existing 3D model. Once the model is fully updated, the digital twin is created, providing a precise digital representation of the patient. The digital twin is then transferred to Blender Software for simulation and analysis of therapeutic exercises. At this stage, the physical therapist performs exercises on the virtual model to evaluate their effectiveness and review the resulting physiological indicators. Finally, the patient undergoes the therapeutic exercises tailored based on the analysis, with the Smart Jacket monitoring the patientâs response during the actual exercises. All physiological indicators and final analysis are displayed to the therapist, allowing them to refine the treatment plan for optimal results.

3.2 Requirement Specification

- Functional Requirements

  - Creating the digital twin based on imaging data and sensor readings.

  - Updating the 3D model with real-time data.

  - Providing comprehensive and accurate analysis for the physical therapist to support treatment decisions.

- Non-Functional Requirements

  - Speed: Achieving fast response times for real-time data updates.

  - Security: Ensuring the protection of patient health data through encryption techniques.

  - Scalability: Supporting a larger number of users or devices in the future without compromising performance.

3.3 Hardware

Smart Jacket Design

The jacket is equipped with advanced sensors, such as ECG sensors and FLEX sensors, which accurately collect and analyze posture data. The primary objective of this jacket is to provide an innovative way to measure pain levels and establish a connection between the physical and digital twin, contributing to the correction of unhealthy postures in real-time.

Moreover, the jacket offers several benefits, including detecting excessive curvature, sending alerts to users when adopting improper postures, and providing instant feedback to help improve body alignment. This innovation represents a significant step toward enhancing back health and reducing problems caused by daily habits.

3.3.1 Hardware Components

Figure 3.2: Clock Sensor Sgw100-2b

Item: Clock Sensor

Quantity: 1

Description: Sgw 100-2b

Approximate Cost: 14-150 SAR

Reason for Use: To measure time or synchronize sensor readings for accurate data.

Figure 3.3: SD Model

Item: SD Model

Quantity: 2

Description: SD Memory-12GB

Approximate Cost: 38-400 SAR

Reason for Use: To store collected data from sensors for later use

Figure 3.4: Electrode for Sensor

Item: Electrode for Sensor

Quantity: 6

Description: HI7609829-10

Approximate Cost: 19-200 SAR

Reason for Use: To sense electrical signals from the body, such as muscle signals.

Figure 3.5: Regulator Source for Battery

Item: Regulator Source for Battery

Quantity: 1

Description: Baku BK 305d

Approximate Cost: 100-600 SAR

Reason for Use: To regulate voltage and maintain current stability

Figure 3.6: Rechargeable Battery

Item: Rechargeable Battery

Quantity: 2

Description: Li-lon 18650 3.7V 2800mAh

Approximate Cost: 19-41 SAR

Reason for Use: To provide a sustainable power source for the system

Figure 3.7: Battery Charger Module

Item: Battery Charger Module

Quantity: 1

Description: Tp4056

Approximate Cost: 8-30 SAR

Reason for Use: To charge the batteries used in the system.

Figure 3.8: Girdle with Aluminum Support

Item: Girdle with Aluminum Support

Quantity: 1

Description: Lumbar Sacral Support

Approximate Cost: 40-188 SAR

Reason for Use: To fix sensors at specific positions on the user's body.

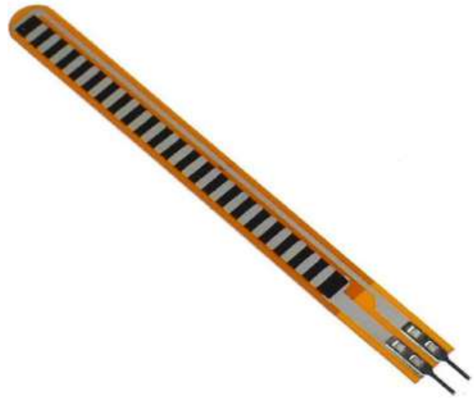


Figure 3.9: Flexion Sensor

Item: Flexion Sensor [29]

Quantity: 6

Description: 2.2 Flex

Approximate Cost: 19-200 SAR

Reason for Use: To accurately measure bending motions in various activities.

Figure 3.10: Protoboard

Item: Protoboard

Quantity: 1

Description: 400 puntos

Approximate Cost: 4-40 SAR

Reason for Use: To temporarily connect electronic components during assembly.

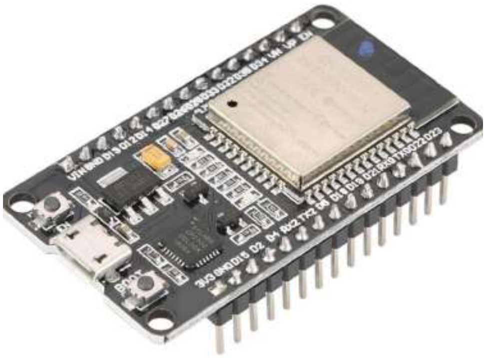


Figure 3.11: ESP32

Item: ESP32 [30]

Quantity: 1

Description: ESP32 Board

Approximate Cost: 19-38 SAR

Reason for Use: For system control and support of wireless communication.

Figure 3.12: ECG Sensor

Item: ECG Sensor [31]

Quantity: 2

Description: AD8232

Approximate Cost: 38-90 SAR

Reason for Use: To monitor and analyze heart signals.

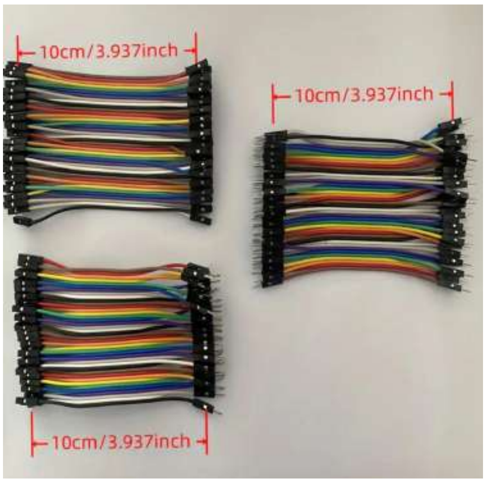


Figure 3.13: Jumper Cable

Item: Jumper Cable

Quantity: 2

Description: 10CM M-M

Approximate Cost: 4-27 SAR

Reason for Use: To quickly connect wires between electronic components.

Figure 3.14: Switch

Item: Switch

Quantity: 6

Description: 2T Rocker Type

Approximate Cost: 4-8 SAR

Reason for Use: For easy on/off control of the system.

Figure 3.15: Mini Push Button

Item: Mini Push Button

Quantity: 2

Description: 6x6x5mm

Approximate Cost: 8-19 SAR

Reason for Use: For manual control of the system functions.

Figure 3.16: Brooch

Item: Brooch

Quantity: 2

Description: T Type for 9V Battery

Approximate Cost: 8-24 SAR

Reason for Use: To connect batteries to other components.

Figure 3.17: Battery

Item: 9V Toshiba Battery

Quantity: 2

Description: 9V Battery

Approximate Cost: 8-20 SAR

Reason for Use: To power the system components.

3.3.2 Schematic Diagram And Connections

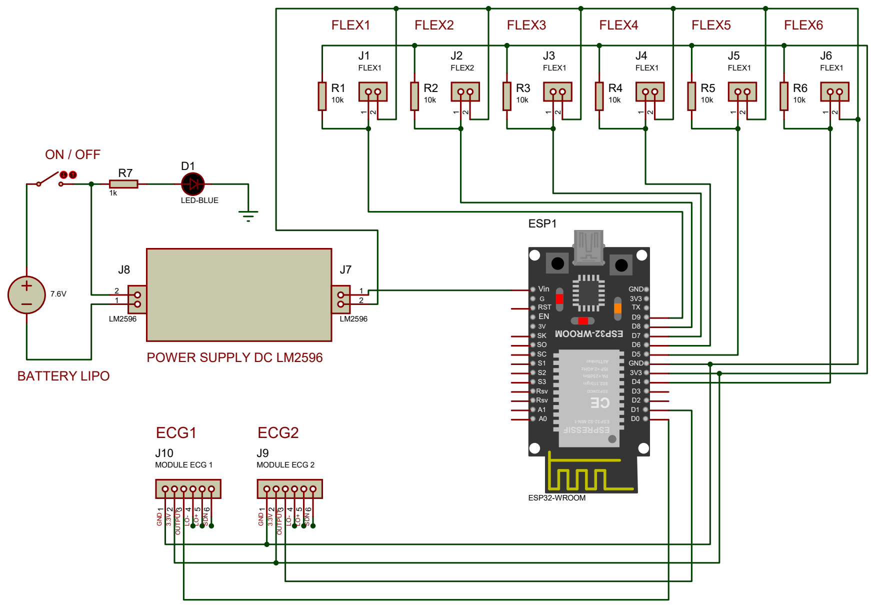


Figure 3.18: Design of the Electronic Scheme Centered on the ESP32 Board

The schematic diagram in Figure 3.18 details the intricate hardware integration of a smart jacket designed to support spinal rehabilitation. The system relies on the ESP32 microcontroller, which serves as the primary processing unit and central communication hub. The ESP32 is connected to various sensors and modules, ensuring seamless data collection, processing, and transmission. Flex sensors are linked to the microcontroller via ports J3 through J8, and ECG modules are connected through J9 and J10, forming the primary data input for the system. These sensors and modules work in tandem to provide comprehensive monitoring. The DS3231 real-time clock module (J1) is included in the configuration to add precise timestamps to the collected data. This ensures the creation of an organized and detailed record of therapy sessions, aiding in the analysis and tracking of rehabilitation progress. To facilitate local storage, the system integrates a Micro SD card module (J2). This module enables the device to save data directly to an external memory card, ensuring reliability even if wireless connectivity is temporarily lost. The device is powered by a rechargeable LiPo battery, which is regulated by two LM2596 voltage regulators (J11 and J12). These regulators stabilize the voltage supply, protecting the system from potential disruptions caused by power fluctuations. Visual feedback is provided by LED indicators (D1 and D2), which are linked to the microcontroller. These LEDs serve to inform the user of the systemâs operational status, such as data recording activity or battery life. Connectivity is further enhanced by the built-in Wi-Fi module of ESP32, which enables wireless data transmission to the digital twin platform. This functionality is critical for integrating the jacket into the digital twin system, allowing real-time analysis and remote monitoring. The physical layout of the circuit is compact and strategically designed to minimize interference and optimize portability. Connections between the sensors, modules, and microcontroller are efficient, ensuring accurate data transmission and robust system performance. By combining precise sensor data acquisition with reliable storage and wireless communication capabilities, the smart jacket offers an effective solution for enhancing spinal rehabilitation outcomes.

3.3.3 Raw Data Acquisition

	Date	Hour	SECG-1	SECG-2	SF1	SF2	SF3	SF4
0	2024-02-27	18:27:02	2.013298	2.889441	47.603106	26.226105	32.594034	36.974173
1	2024-02-27	18:27:03	0.203440	0.737184	51.771432	46.088802	49.984907	49.450889
2	2024-02-27	18:27:04	1.983038	1.547115	21.784425	24.632529	23.965613	27.344828
3	2024-02-27	18:27:06	3.289107	2.867787	40.089411	42.968482	57.668563	49.225209
4	2024-02-27	18:27:07	1.621162	1.793106	57.352814	58.876270	59.260652	53.502097
...	...	...	...	...	...	...	...	...
54140	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118
54141	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118
54142	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118
54143	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118
54144	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118

54145 rows × 9 columns

Figure 3.19: Example of Expected Data

Figure 3.19 presents sample raw data output from the system, including digital values from the flex sensors and waveform data from the ECG modules. This figure has been adapted from a reference similar to the smart jacket's design [32] to illustrate the type of data our system captures.

Each sensor in the system outputs raw data signals that are processed by the ESP32 microcontroller. The flex sensors (J3 - J8) generate analog voltage signals that correspond to the degree of spinal bending. These signals are converted into digital values using the ESP32 ADC (Analog-to-Digital Converter), resulting in numerical values ranging from 0 to 4095. These values reflect the intensity of bending and form the basis for further analysis. Similarly, the ECG modules (J9 and J10) produce analog signals representing electrical activity from back muscles. These signals are amplified, filtered, and then digitized by the ESP32. The raw ECG data is typically represented as voltage values within a range of millivolts, depending on the system configuration, and is sampled at predefined intervals, generating waveforms that reflect muscle activity. The DS3231 real-time clock module (J1) provides precise time-stamping for all sensor readings, ensuring the collected data is organized in a meaningful way. The Micro SD card module (J2) stores these raw data outputs, along with their corresponding timestamps, in structured log files, guaranteeing data persistence even if wireless connectivity is temporarily lost.

These raw data visualizations in Figure 3.18 are crucial for understanding the initial measurements and accurately analyzing rehabilitation progress.

3.4 Software

In this section, we will focus on raw data to estimate pain levels and to establish a connection between the physical and digital twin. Raw data refers to the primary information collected directly from sensors or devices before any processing or cleaning is applied. This data is stored in a database to facilitate easy access and later analysis. Table 3.1 provides a practical example of the structure of this data, helping to understand the type of information coming from various sensors related to the physiological and motion indicators collected by multiple sensors [32].

Table 3.1: Raw Data

	Date	Hour	SECG-1	SECG-2	SF1	SF2	SF3	SF4	Pain
0	2024-02-27	18:27:02	2.013298	2.889441	47.603106	26.226105	32.594034	36.974173	1.0
1	2024-02-27	18:27:03	0.203440	0.737184	51.771432	46.088802	49.984907	49.450889	3.0
2	2024-02-27	18:27:04	1.983038	1.547115	21.784425	24.632529	23.965613	27.344828	1.0
3	2024-02-27	18:27:06	3.289107	2.867787	40.089411	42.968482	57.668563	49.225209	3.0
4	2024-02-27	18:27:07	1.621162	1.793106	57.352814	58.876270	59.260652	53.502097	3.0
...	...	...	...	...	...	...	...	...	...
54140	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118	2.0
54141	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118	2.0
54142	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118	2.0
54143	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118	2.0
54144	2024-03-05	20:35:14	1.881215	1.819028	37.103334	37.244623	31.134827	30.184118	2.0

54145 rows × 9 columns

Below is an explanation of the key concepts listed in the table and the sources of the corresponding data:

Date: Indicates the date when the data was collected, aiding in the analysis of temporal patterns associated with specific days. Source: Clock sensor.

Time: Displays the time of data collection (hour and minute) to track physiological or motion changes within a specific time frame.Source:Clock sensor

SECG-1 - SECG-2: Record the electrical values of cardiac activity.Source:ECG sensor

SF1 - SF4: Represent the angles or curvatures in different parts of the spine. Source: Flex sensor

Pain: Reflects the level of pain recorded based on the patient's movement and muscle activity

3.4.1 Raw Data Utilization

The significance of raw data lies in its critical role in accomplishing two main tasks. The first task involves accurately Estimating the pain level based on the received data, enhancing evaluation and medical intervention. The second task focuses on updating the digital twin by transmitting this data to continuously improve its accuracy and ensure alignment with the patientâs real condition.

Pain Level Estimation

Demonstrates how to process raw data to estimate the pain level using a neural network is essential part. Neural network is a significant approach to integration of Multi-Source Data: Deep Neural Networks are capable of handling heterogeneous data types, such as analog and digital signals, collected from ECG and Flex sensors alongside temporal data, Detection of Nonlinear Patterns: Pain is influenced by complex interactions among physiological and biomechanical factors. Neural Networks excel at identifying these non-linear relationships, Temporal Data Analysis: If the sensor data includes time-sequential patterns (e.g., signal changes over time), models like Long Short-Term Memory (LSTM) or Recurrent Neural Networks (RNN) can effectively analyze temporal trends, Learning from Historical Data: The network can be trained on historical labeled data to predict pain levels for new, unseen data.

Neural Networks are highly effective tools for estimating pain levels due to their ability to process complex and multi-source data collected from various sensors.

In terms of the aforementioned sensor inputs, this research dealing with four flex sensors (SF1, SF2, SF3, SF4) which are responsible to Measure the degree of bending or motion in muscles located in different regions of the lower back. Also, The values are represented as numerical data indicating the intensity of bending or movement. ECG (SECG-1, SECG-2) which are responsible to Measure the electrical activity of muscles. Also, Produce continuous numerical signals representing the intensity of electrical activity.

The network's outputs indicate the anticipated pain level, classified into three categories: Low Pain, Moderate Pain, and High Pain. It generates probabilities for each class, where the highest probability determines the final classification. For example, an output of [0.1, 0.2, 0.7] indicates a 10 percent probability for Low Pain, 20 percent for Moderate Pain, and 70 percent for High Pain. In this case, the final pain level classification would be High Pain, as it has the highest probability.

CNN Approach Design

The Figure 3.20 illustrates the structure of a Convolution-al Neural Network (CNN) designed to process and analyze multi-modal data to predict pain level prediction. The inputs include ECG Data (heart rate variability), Flex Sensor Data (movement patterns), and temporal data (time stamps). These inputs go through a pre-processing phase before being analyzed by specific layers, dense neural networks process independent data, LSTM/RNN layers process sequential data, and CNN layers extract spatial features. The network outputs two types of predictions: classification of pain levels (Low, Moedium, High) and regression for numerical pain levels (scale 1-10).

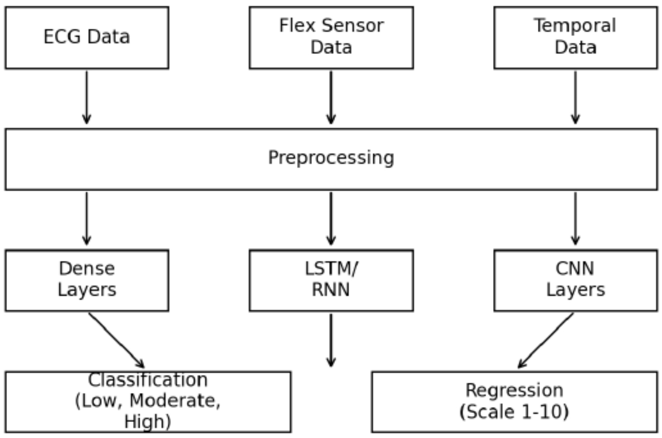


Figure 3.20: CNN Structure for Pain Prediction From Multi-Modal Data

3.4.2 Creating Digital Twin

To create the digital twin, two main data sources are used: the first comes from medical imaging techniques, which include Bone Scan or CT scan with IV contrast, while the second is data obtained from sensors. After merging these two data inputs, a precise 3D model is generated that reflects the patient's actual condition.

1. **CT Reconstruction Techniques:**

   In order to feed the digital twin with 3D model, the system requires either Bone Scan or CT scan with IV contrast . Bone scan produce high resolution of 3D model required. However, CT scan can either produce 3D model which is not available in all machines or can produce 2D model which needs to go through several steps and processes to be converted into 3D Model.

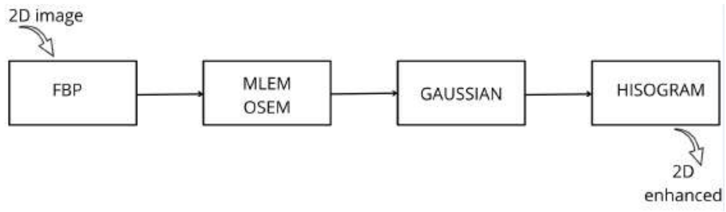


Figure 3.21: Structure of a Radiographic Imaging Procedure

These steps are shown in Figure 3.21 such as first step where is during imaging data collection, Issuing X-rays that rotate around the patient and using the ASIR algorithm to improve quality and reduce dose. The second step where is after imaging the image will be processed using the following algorithms, FBP algorithm for image reconstruction, which is then enhanced via MLEM/OSEM algorithms, Noise reduction using the Gaussian Filtering algorithm, Contrast enhancement using the Histogram Equalization algorithm, After the image passes through the mentioned algorithms and is improved to achieve good quality, a two-dimensional image is produced [33].

Now, the 2D Ct model is ready to be converted into 3D model following the algorithm below:

- Methodology for 3D Reconstruction:

To create a 3D model from imaging data, several essential steps are taken into consideration as shown in Figure 3.22 as follows:

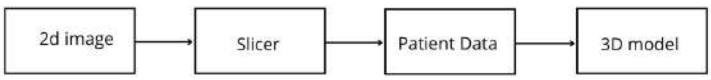


Figure 3.22: Convert From 2D to 3D

(a) Two-dimensional X-ray images are taken.

(b) The images are uploaded to the Slicer program.

(c) Patient data is entered, such as:

i. Height

ii. Weight

iii. Body Mass Index (BMI)

(d) A 3D model is printed based on the entered data.

(e) Save the model in the format: FBX

2. **Bones Scan:**

A whole-body bone scan is a type of nuclear medicine imaging test. Healthcare providers use advanced technology that may seem complicated. Before the test, you receive an injection of a radiotracer (a substance that contains a small, safe amount of radioactive material). The radiotracer collects in areas of your bones where there are changes or irregularities. These areas appear clearly on the imaging scan, helping your healthcare provider pinpoint any disease.

What Does a Whole-Body Bone Scan Show?A whole-body bone scan can reveal damage or changes in your bones. [34] The results of this test are saved in DICOM format, which can be converted into 3D models in either STL or FBX formats. And This will be the method employed.

3. **Updating the Digital Twin Based on Raw Data:**

The Bone Scan test is performed once for the patient initially to map the basic skeletal structure and build an accurate 3D model. After this process, raw data is used during each subsequent visit to update the patientâs digital version between each visit. This continuous update allows for tracking improvements and developments in the digital model, enabling the creation of a new digital version that accurately aligns with the patientâs physical version at each visit. This ongoing process is essential to ensure continuous alignment between the digital twin and the actual condition of the patient, enhancing the accuracy of medical assessments and supporting informed medical decision-making based on up-to-date and realistic data.

3.4.3 Exerisce Platform- 3D model Training Exersice

Now that the 3D model is ready, it can be exported to various development and animation environments for training and performing exercises, with different options for these environments having been thoroughly studied.

After we conducted extensive research on Blender, Unity, and Mixamo, this paragraph will discuss the advantages and disadvantages of each program to provide a comprehensive overview that helps in selecting the most suitable one based on specific needs.

Mixamo: is a specialized platform that utilizes artificial intelligence to simplify the animation of 3D characters. It is characterized by its ease of use and a vast library of pre-made animations that save time. However, it suffers from limitations in customizing animations and controlling fine details [35] [36] [37].

Unity: is a game engine used for developing 2D and 3D games. It provides an integrated development environment and powerful graphical tools to create realistic designs. However, it can be resource-intensive and heavy during operation, requiring high-performance devices to run efficiently [38].

Blender:it is an open-source software that provides comprehensive tools for 3D modeling and animation. It is highly flexible and making character animation more realistic. However, requiring some time to adapt to all the available features [39].

Furthermore, different environments have different technique and mechanism as show below in Table 3.2

Table 3.2: Different Environments Uses Different Algorithm

Programs	Blender	Unity	Mixamo
Background algorithm for animation	iTasC Solver (Inverse Kinematics Task Controller) Physics-Based Inverse Kinematics	Keyframe Animation Animation Controllers Blend Trees Skeletal Animation Procedural Animation Inverse Kinematics (IK)	Auto Rigging Motion Retargeting Linear Blend Skinning (LBS) Pre-Baked Animation Pose Interpolation

We conclude our summary of the comparison between three environments in terms of rigging , library of pre-rigged 3D character model , animation design and exports files in Table 3.3:

Table 3.3: Comparison Between Environments

Feature	Blender	unity	Mixamo
Auto Rigging	✗	✗	✓
User-defined rigging	✓	✓	✗
Library of pre-rigged 3D character model	✗	✗	✓
Easy to use in animation design	✗	✗	✓
Exports files in multiple formats such as FBX and BAE	✓	✓	✓

Mixamo is the ideal choice for our project as it is a web-based tool, which allowed us to integrate it directly into our system without requiring any installation on the user device. This made it faster to implement, easier to use, and more adaptable to different hardware setups.

Chapter 4 | Practical Implementation and results

4.1 System Workflow During Practical Use

The previous chapter focused on the process by which the data collected from the sensors is transmitted, received, and stored in the database to ensure continuous updates of the patient's digital twin. Building on that, this section provides a detailed explanation of the system workflow during real-world use. Each user interface is examined to illustrate how users interact with the system in practice. The focus is not only on the visible elements of interaction, but also on the underlying processes that occur in the background. By breaking down each interface, we aim to clarify how input is received, processed, and transformed into meaningful output. This helps in understanding both the functional behavior and the technical mechanisms that support the user experience.

4.1.1 System Interfaces and Their Interactions

1. Welcome Screen:

Upon launching the system, a splash screen appears for 5 seconds before automatically transitioning to the login interface.

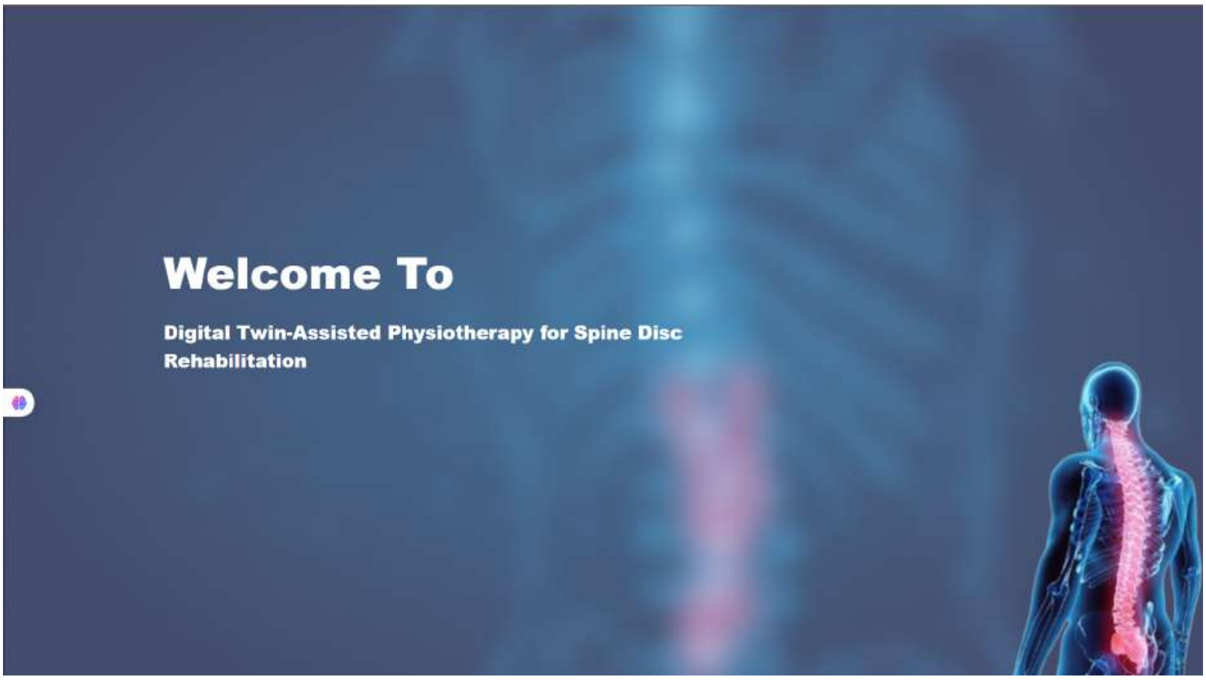


Figure 4.1: Interface 1 - Welcome Screen

2. Login Screen:

Employees log in using their credentials. The system verifies the entered data against the Users table in the database:

- If the credentials are valid, access is granted.
- If invalid, an error message is displayed.

There is also an option to create a new account for newly hired employees.

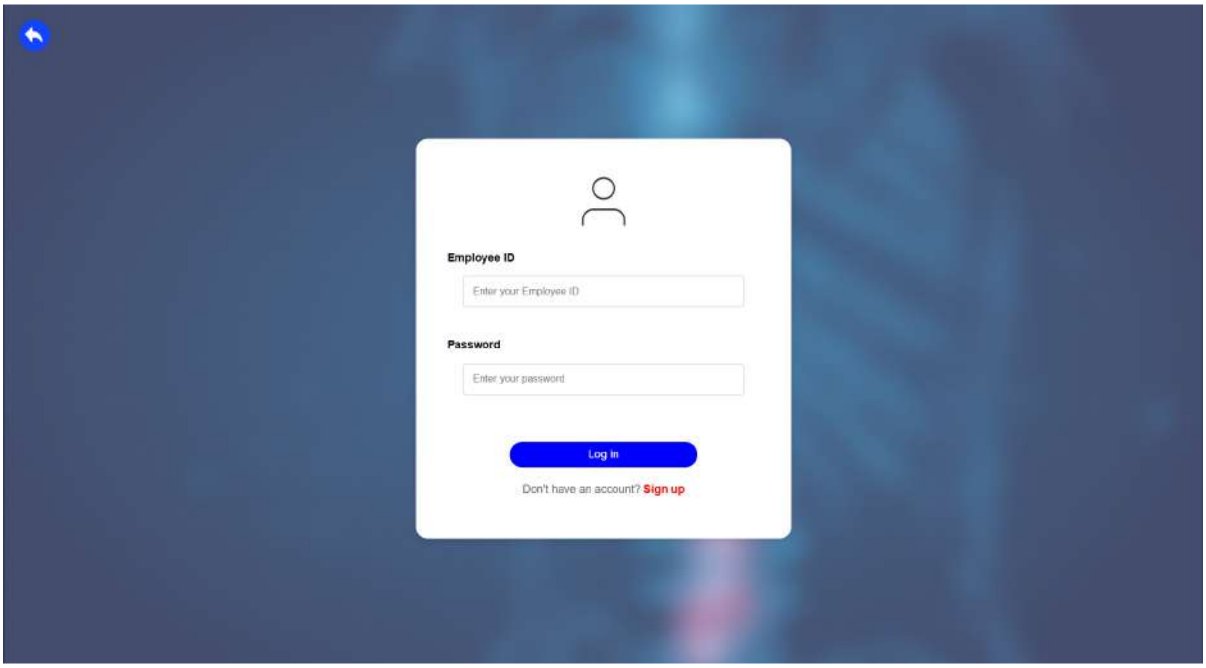


Figure 4.2: Interface 2 - Login Screen

3. Role Selection Interface:

After successful login, the user is prompted to identify their role as either a Nurse or a Physiotherapist:

- If (**Nurse**) is selected, the system checks the Nurses table for user data.
- If (**Physiotherapist**) is selected, the Physiotherapist table is checked.
- If no matching record is found, an error message is displayed.

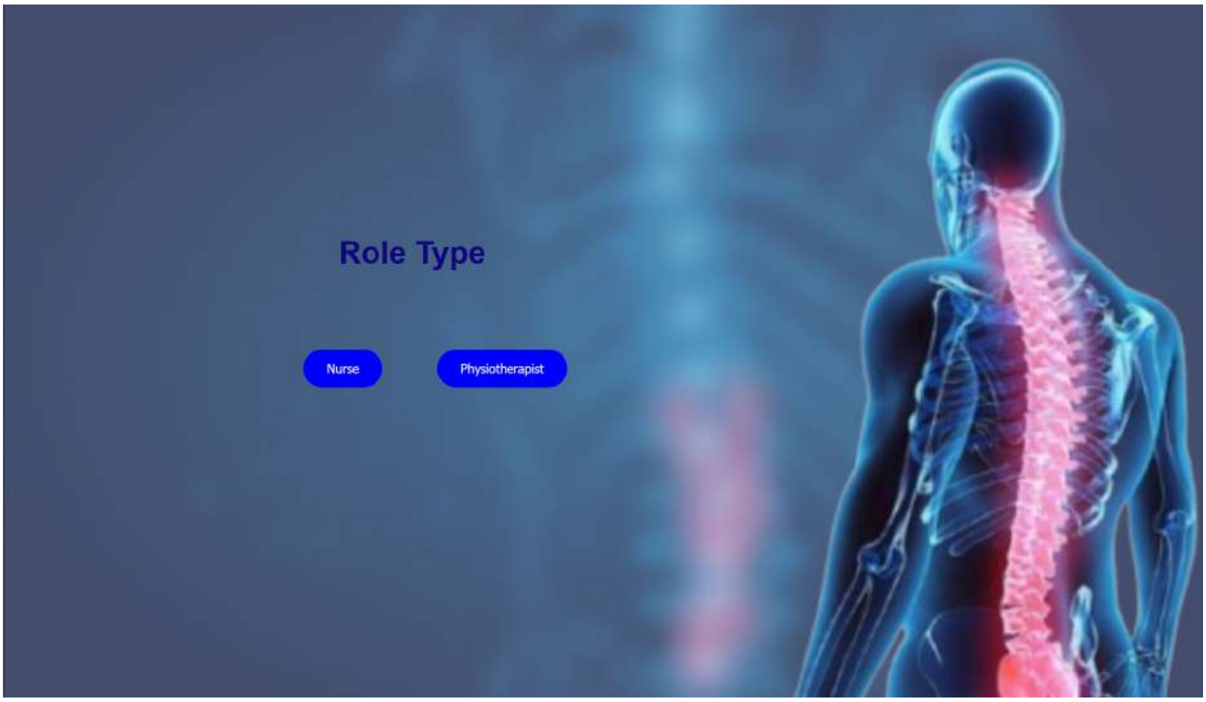


Figure 4.3: Interface 3 - Role Selection Interface

Nurse Workflow:

4. Patient Status Selection:

The nurse selects whether the patient is new or already registered:

- If the patient is new, the system navigates to **Interface** 4.5 for patient registration.
- If the patient already exists, **Interface** 4.6 is displayed showing a list of registered patients.

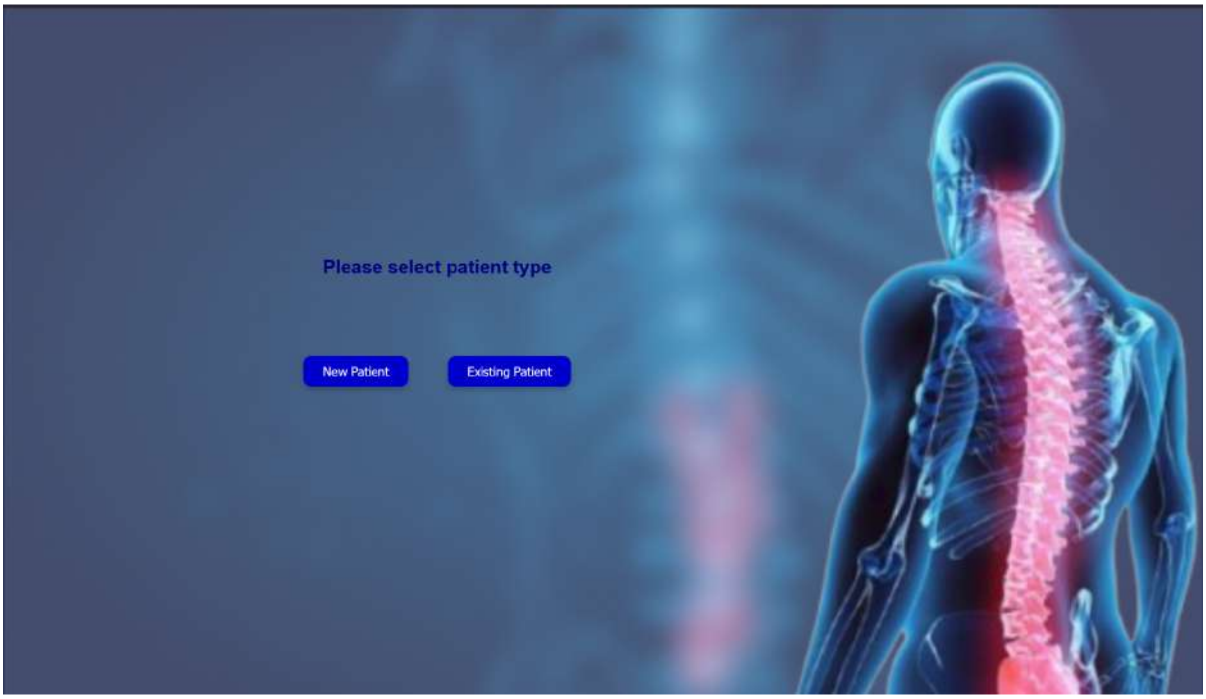


Figure 4.4: Interface 4 - Patient Status Selection

5. New Patient Registration:

In this interface, the patientâs data is entered and stored across the following database tables:

- **Patients**: includes basic information (name, age, gender).
- **Contact Information**: stores the phone number, email, and address.
- **Medical Information**: contains medical history, chronic conditions, etc.

In the **Import X-ray file** field, a 3D X-ray file obtained and processed by the radiology department is attached and stored in the **Scan Data** table.

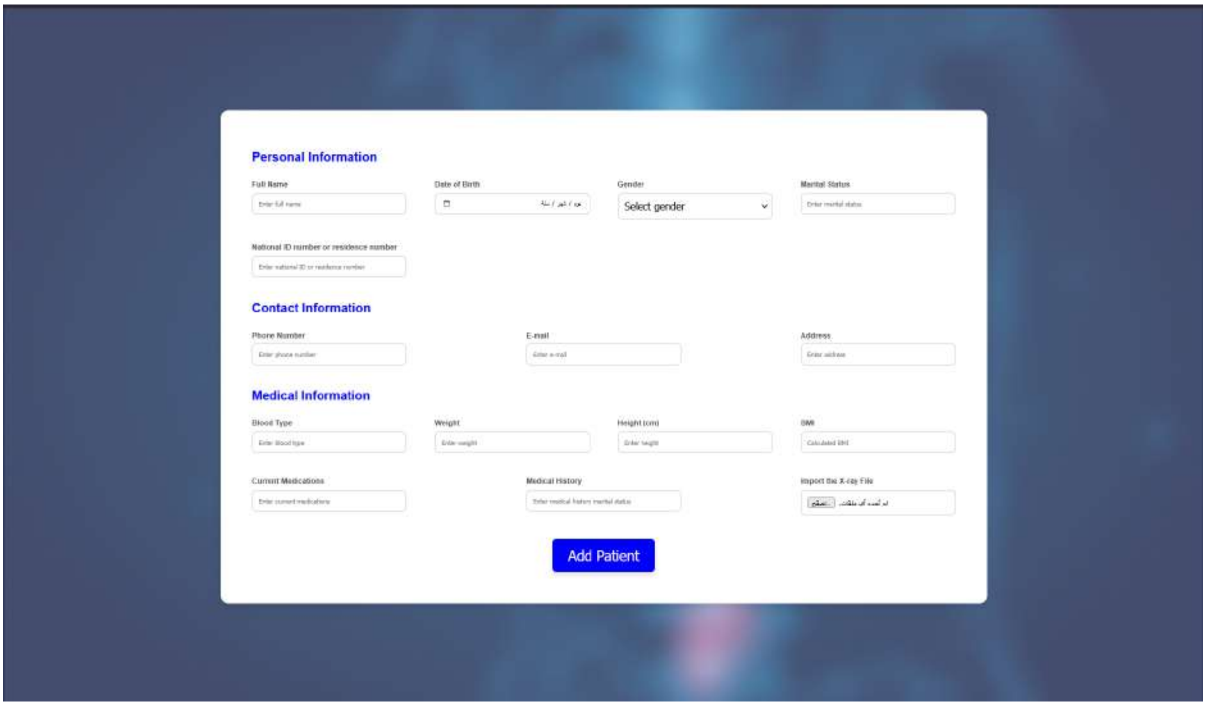


Figure 4.5: Interface 5 - New Patient Registration

6. Home Page:

Displays a list of all patients with the following details per patient: Patient ID ,Full Name ,X-ray Status (Processed or Not) ,Diagnosis and Current Condition ,Assigned Physiotherapist ,Date of ,Last Session

Each patient entry includes a three-dot menu with the following options:

- **Open Patient File** : to either edit details or begin a session
- **Show Digital Twin**: to allow physiotherapists to view the 3D twin and pain level
- **Delete File**: to remove the patient's record if their treatment has concluded

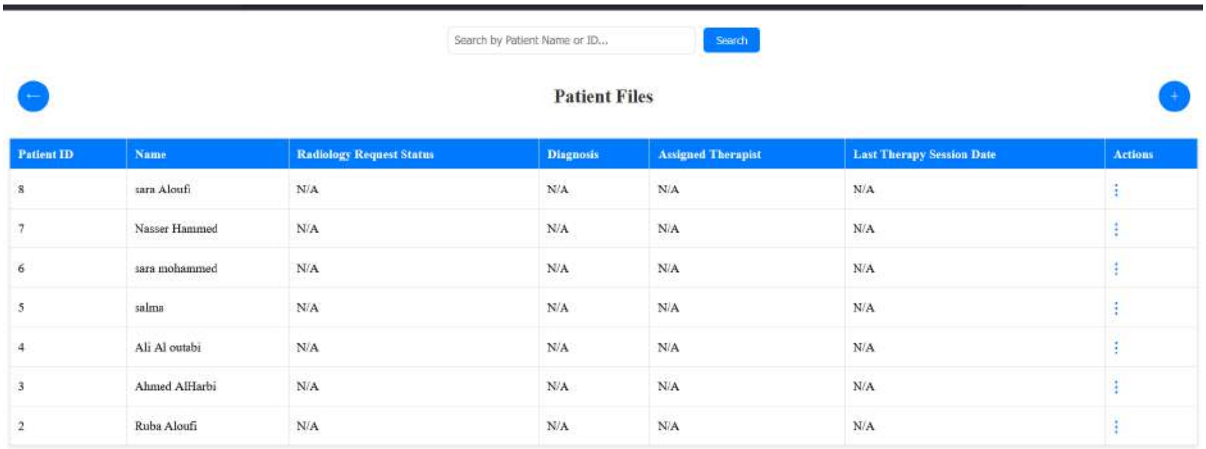


Figure 4.6: Interface 6 - Home Page

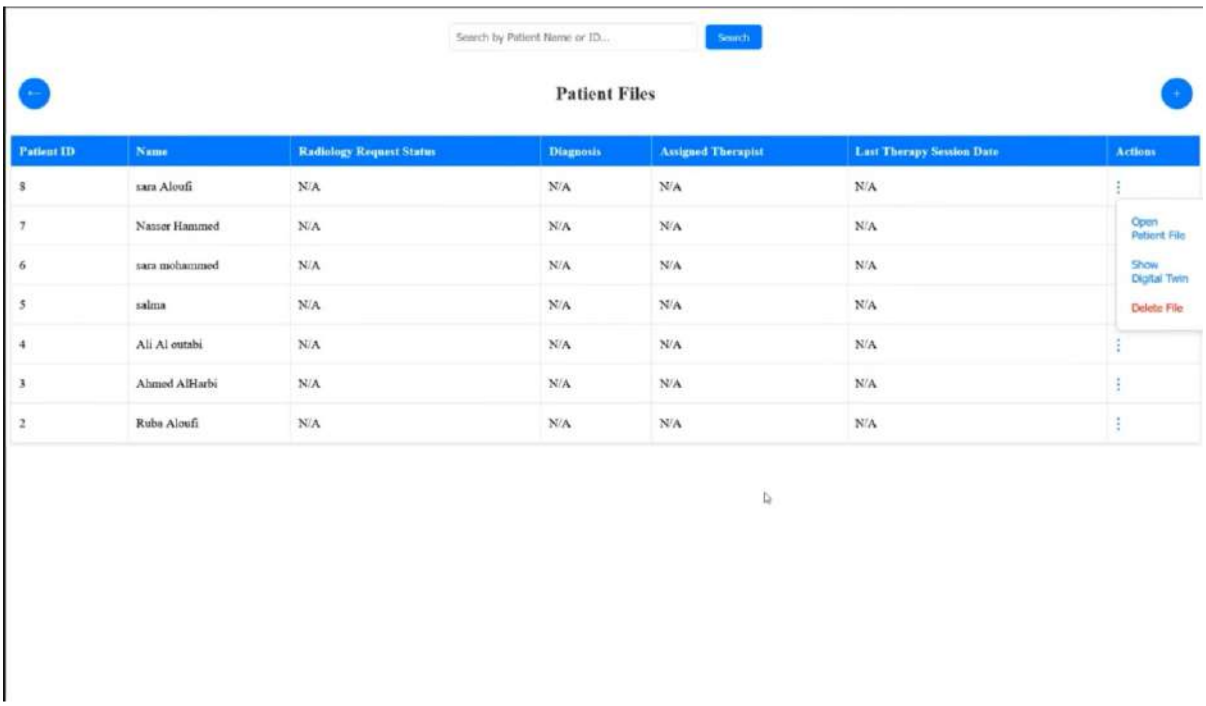


Figure 4.7: Interface 7 - Home Page (options)

7. Edit Patient Data or Start a Session:

From this interface, the nurse can:

- Modify the patientâs information **Interface** 4.5
- Begin a session by entering the patientâs vital signs **Interface** 4.8

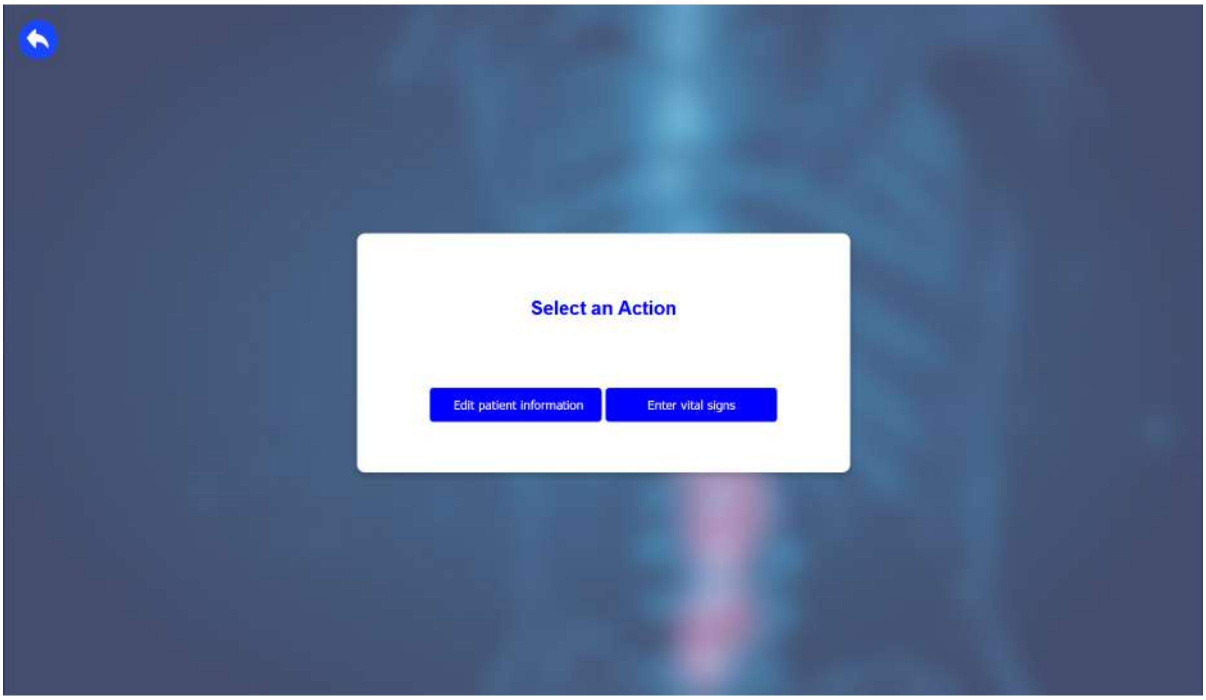


Figure 4.8: Interface 8 - Edit Patient Data or Start a Session

8. Vital Signs Input:

The patient's vital signs are recorded and saved in the Vital Signs table. Upon clicking Update File, the data is stored and the system transitions to Interface 4.9

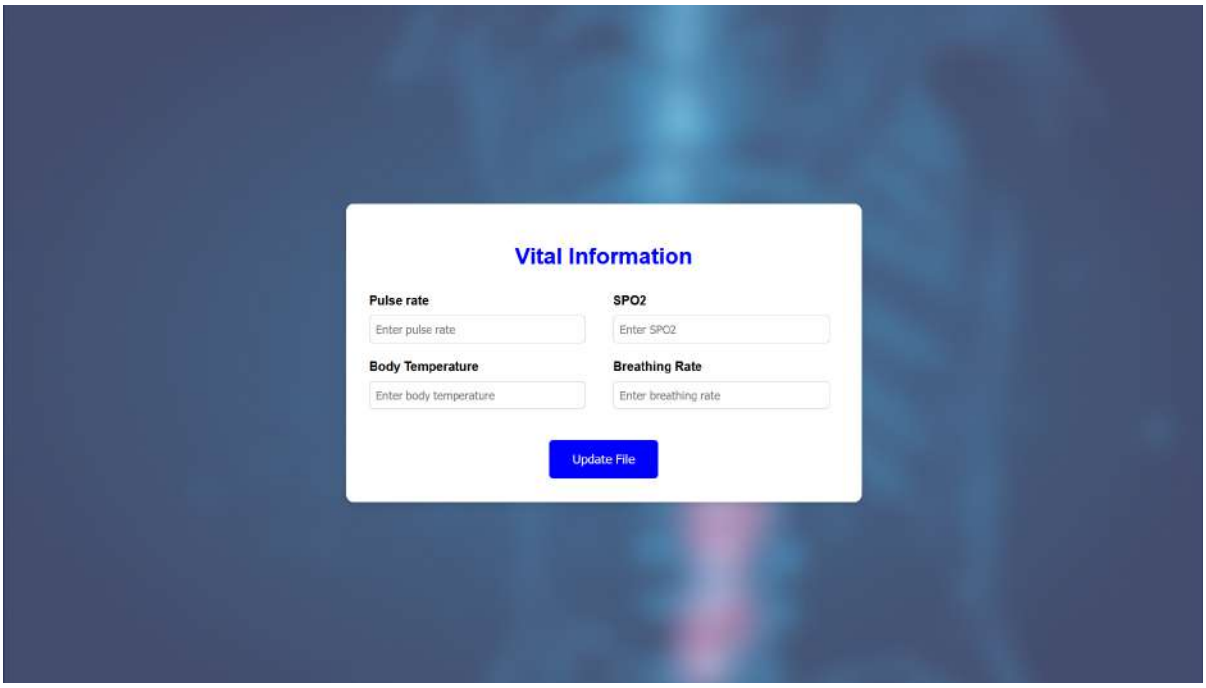


Figure 4.9: Interface 9 - Vital Signs Input

9. Patient Readiness Evaluation:

- If the patient is deemed fit for a session, **Start the Session** is clicked, and the smart vest is worn to collect real-time data **Interface 4.10**
- If the patient is unfit, the system returns to the home page.

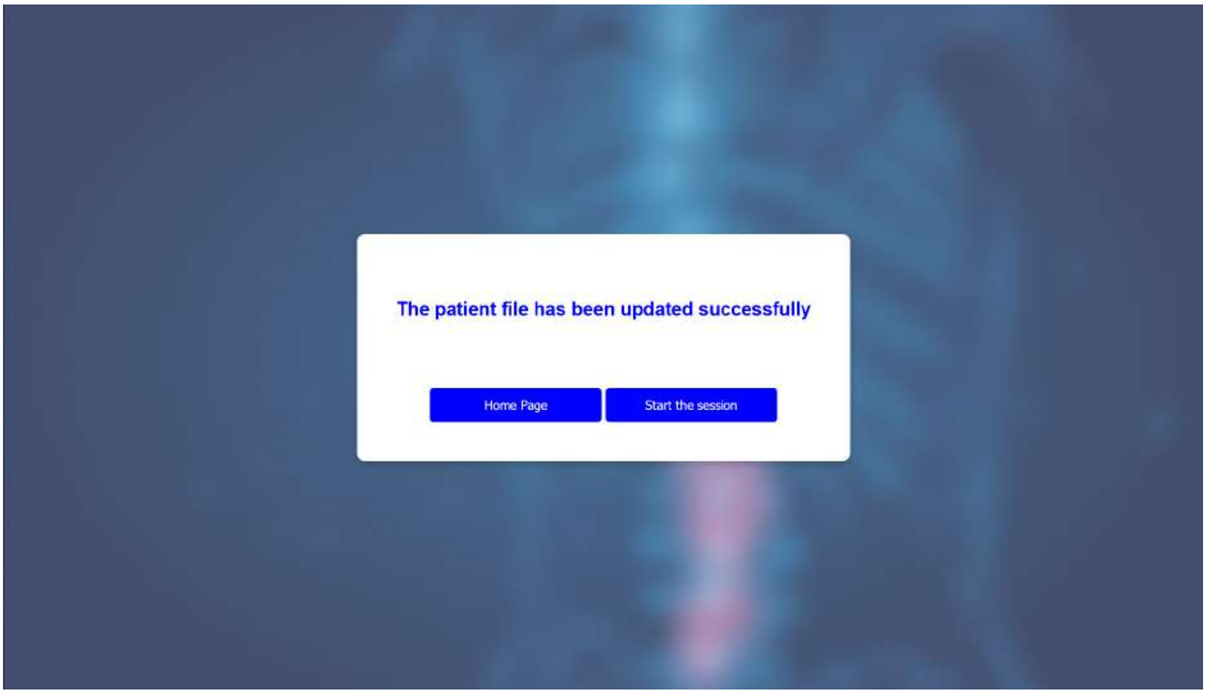


Figure 4.10: Interface 10 - Patient Readiness Evaluation

10. Smart Vest Data Collection:

Vital measurements collected via the smart vest are uploaded and stored in the Smart Vest table. After data upload is complete, the system moves to Interface 4.12

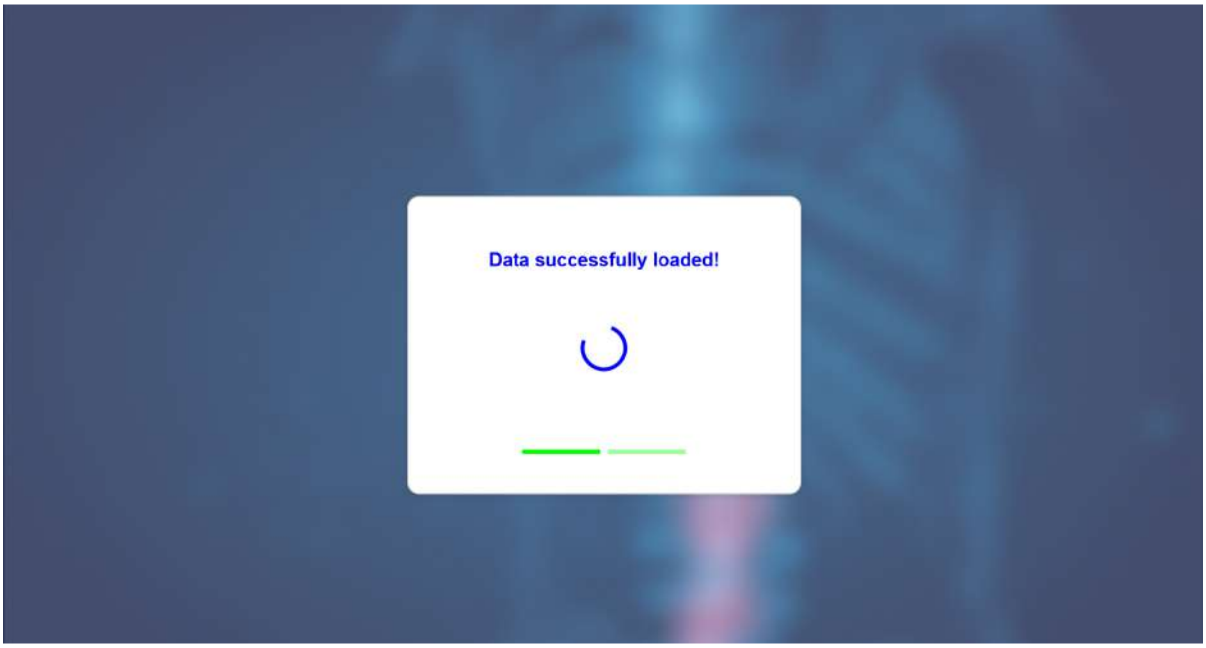


Figure 4.11: Interface 11 - Smart Vest Data Collection

11. Digital Twin Confirmation:

A confirmation message indicates that the patientâs digital twin has been successfully generated. Clicking Return to Home Page navigates back to the main interface.

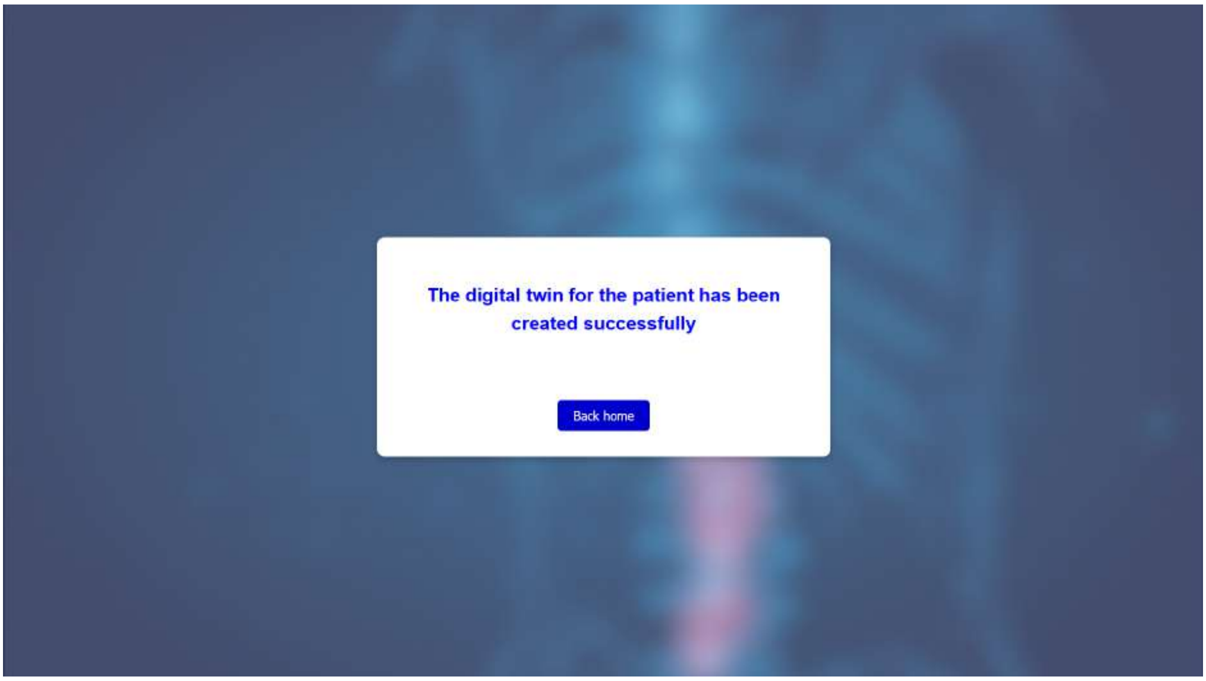


Figure 4.12: Interface 12 - Digital Twin Confirmation

Physiotherapist Workflow:

After verifying the physiotherapist's credentials, the system displays the home page containing the list of patients. Next to each patient entry, a menu includes **Show Digital Twin** , which allows the physiotherapist to view the full digital twin model.

12. Digital Twin Viewer Interface 4.13:

This interface consists of the following components:

- **Vital Signs**: Displays the following values retrieved from the **Vital Signs** and **Medical Information** tables:
  - Pulse Rate
  - Oxygen Saturation (SPO2)

  - Breathing Rate
  - Body Temperature
  - Body Mass Index (BMI)

- **Pain Level**: The pain level is calculated using data collected from the **Smart Vest** table. This data is sent to a Flask server, which integrates with a Convolutional Neural Network (CNN) model to compute pain level predictions.

- **3D Digital Twin Model**: Displayed on the right side of the screen, retrieved from the **Scan Data** table. Clicking the **Export 3D Model** button downloads the model to the desktop and simultaneously, the **Exercises button** is activated, allowing the user to navigate to the therapeutic exercise setup interface using the digital model.

Figure 4.13: Interface 13 - Digital Twin Viewer Interface

Exercises button 4.14: Upon activation, the Exercises button redirects the user to the Mixamo platform where the digital twin model is uploaded for animation processing 4.15. Once the model is loaded, specific joints can be selected to configure targeted therapeutic movements 4.16 .After selecting the targeted joints, an appropriate motion is chosen from the predefined **Sport Exercises** library. The digital twin then performs the selected therapeutic exercise. Figures 4.17 and 4.18 depict the same therapeutic movement from different viewing angles, offering a more comprehensive and spatial understanding of the exercise execution.

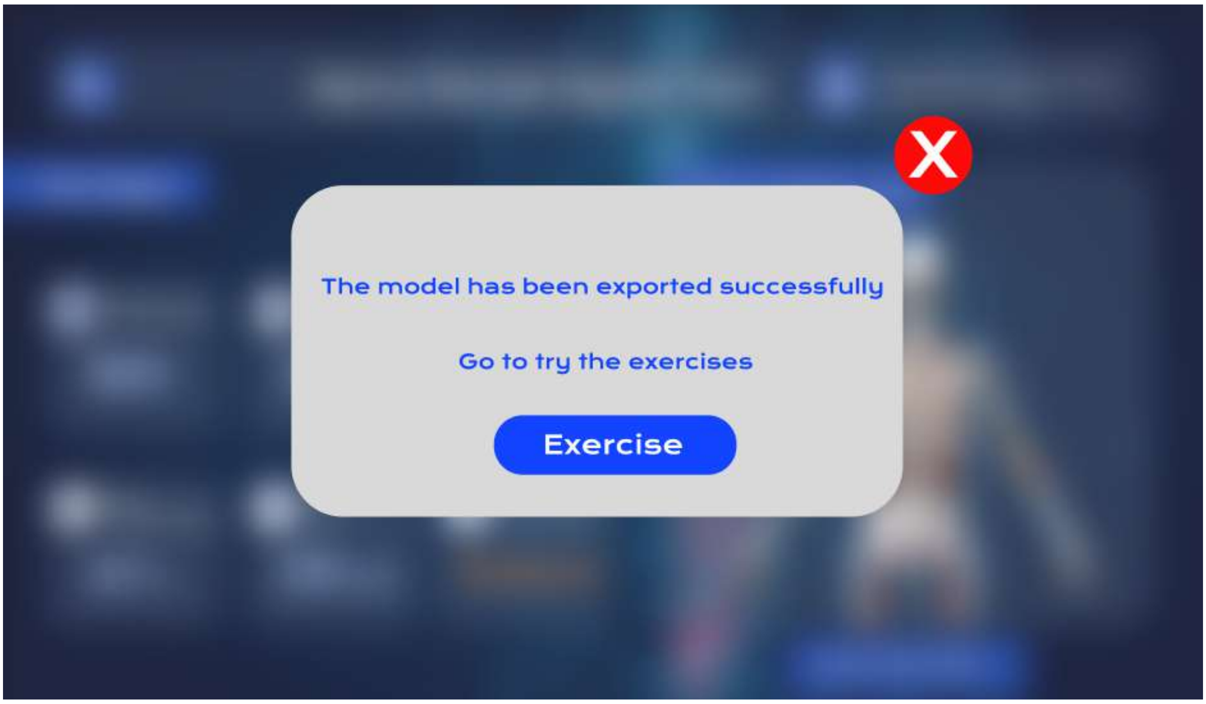


Figure 4.14: Interface 14 - Exercises Button

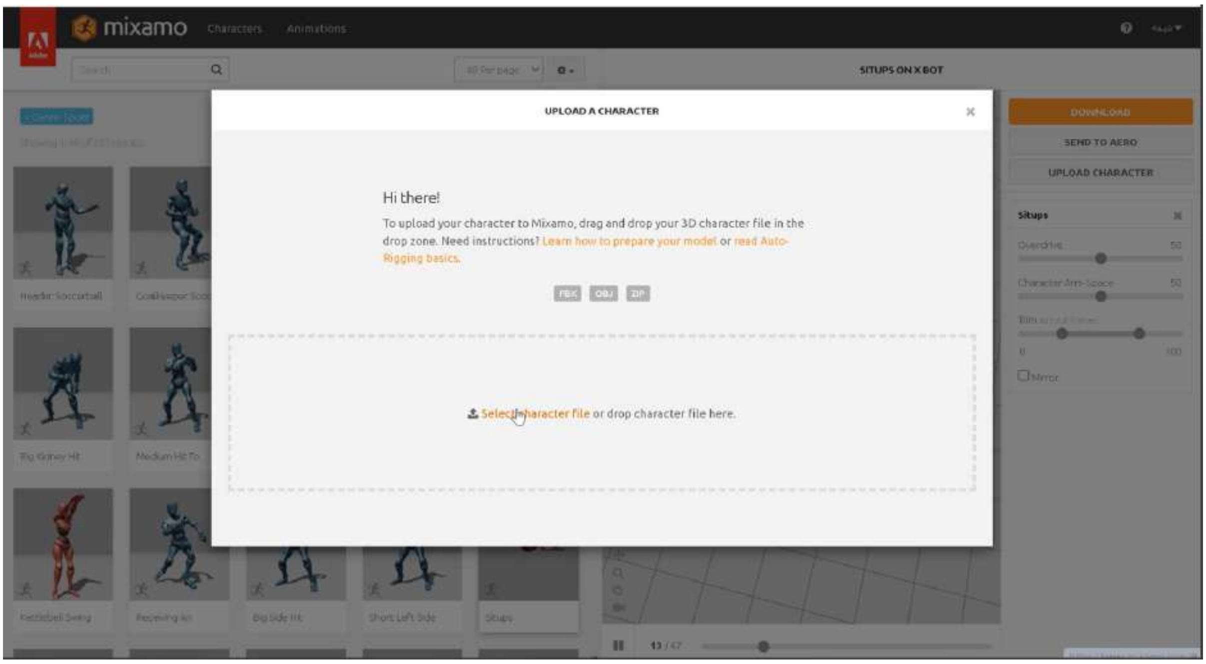


Figure 4.15: Interface 15 - Uploading the 3D Model of the Patient

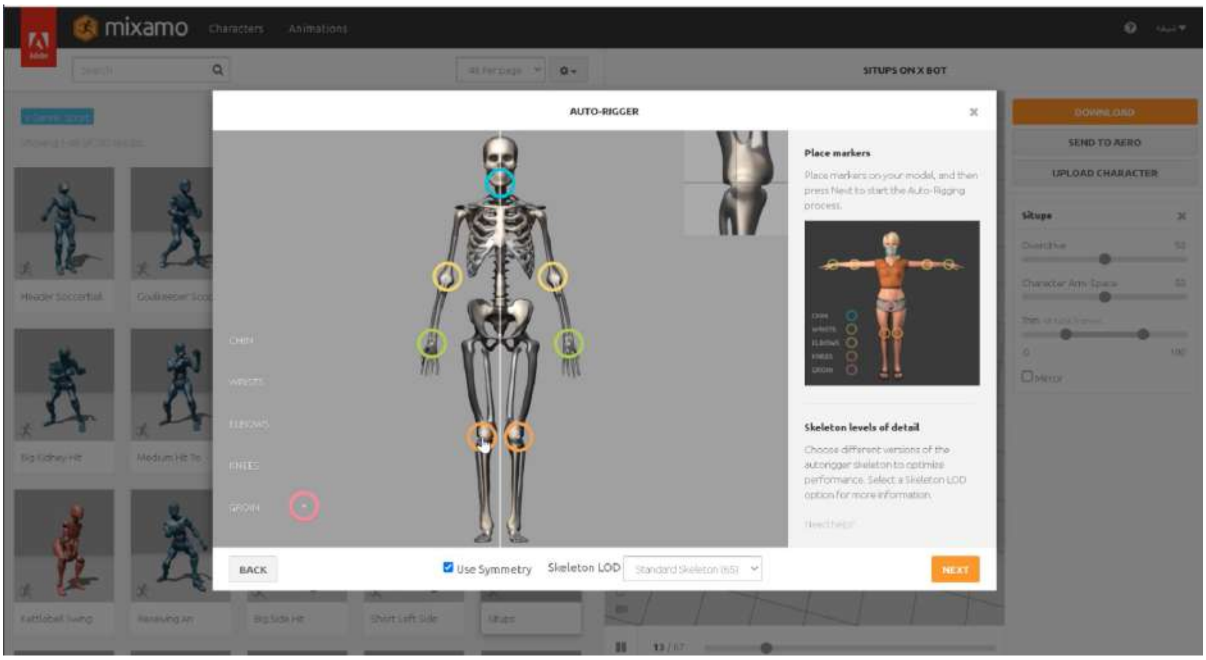


Figure 4.16: Interface 16 - Rigging the Model

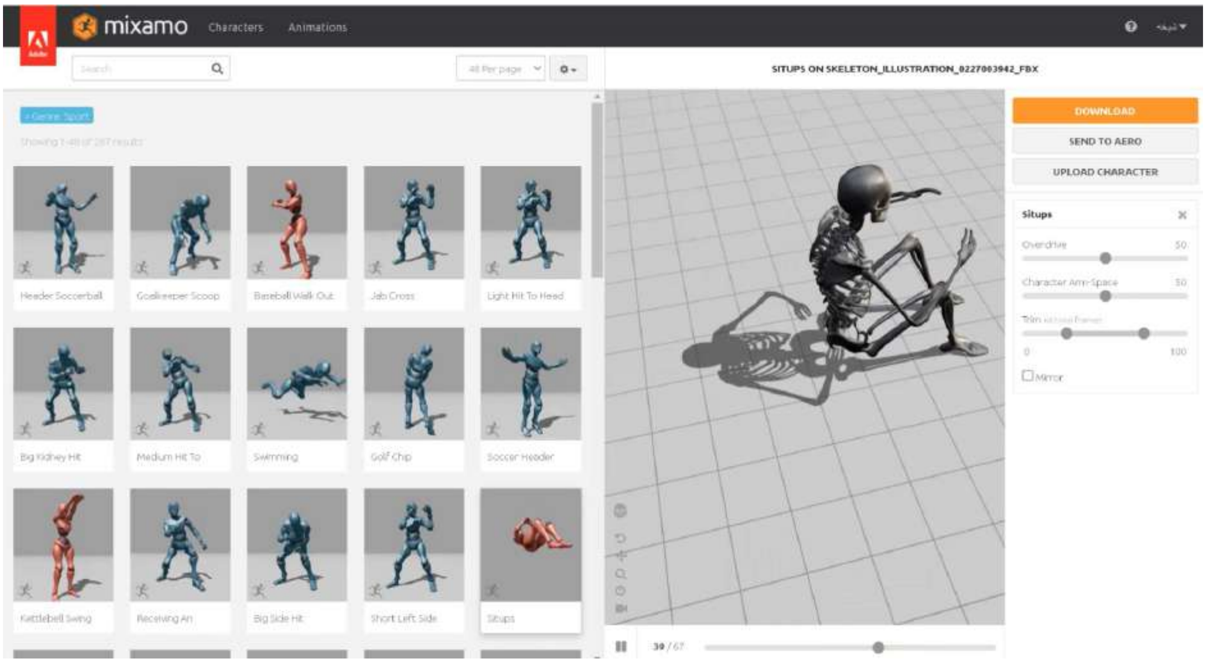


Figure 4.17: Interface 17 - The rig performs Exercise A, as seen from the right-side perspective, demonstrating the movement of the skeleton

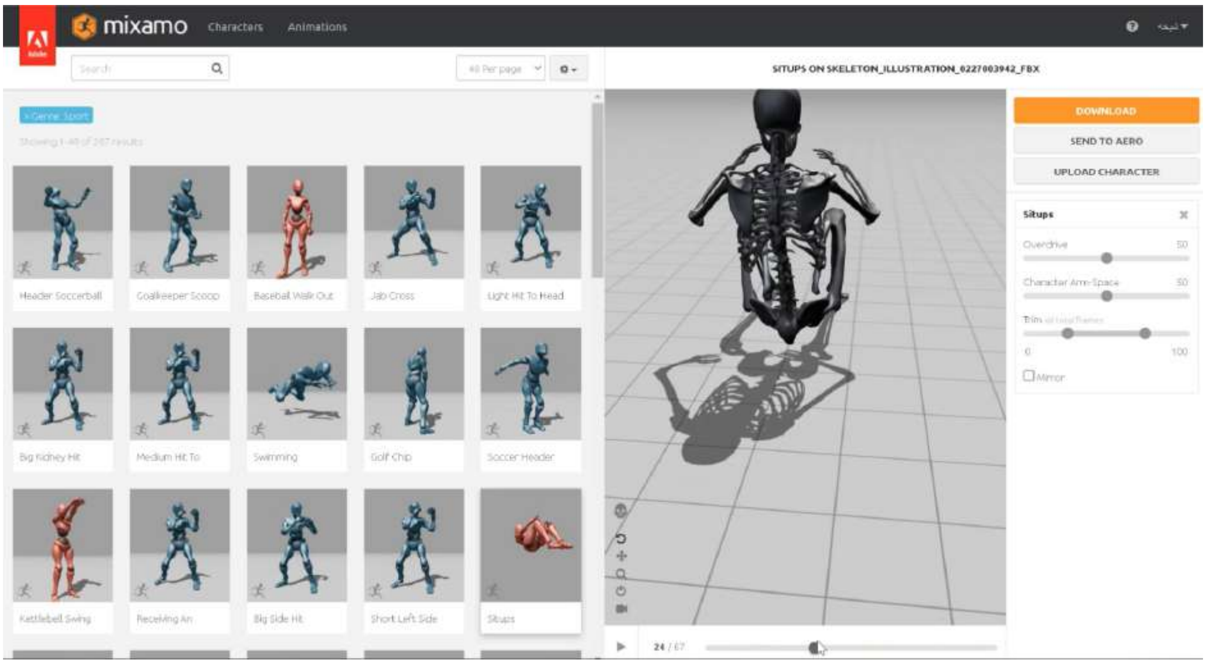


Figure 4.18: Interface 18 - The Rig Performs Exercise A,as seen from the rear perspective, demonstrating the movement of the skeleton

4.2 Programming and Model Development

4.2.1 Development Environment Setup and Component Integration

The system development employed an integrated environment encompassing specific hardware components and software platforms. The data flow architecture was designed to facilitate real-time pain assessment through the analysis of multimodal physiological signals. This holistic approach ensures seamless operation and efficient data processing for accurate and timely pain evaluation, providing a robust foundation for the research.

Hardware Platform and Sensor Configuration

The ESP32 NodeMCU serves as the central processing unit for the wearable system, chosen for its dual-core architecture, integrated Wi-Fi and Bluetooth functionalities, and general-purpose input/output (GPIO) versatility. The smart vest incorporates six flex sensors (SF1–SF6), strategically positioned longitudinally along the spine, to capture data pertaining to bending and curvature. Additionally, two AD8232 electrocardiogram (ECG) sensors are placed on the chest to monitor electrical cardiac activity. These analog signals are directed through an ADS1115 16-bit analog-to-digital converter, thereby augmenting the resolution and precision of the measurements, particularly essential for the accurate acquisition of ECG signals As described in the Scikit-learn documentation [40].

Power and Communication Modules

The device operates on a 3.7V Li-Ion rechargeable battery, the output of which is regulated by an LM2596 voltage regulator. A TP4056 charging module manages the charging process, ensuring a stable 3.3V power supply. Wireless data transmission between the ESP32 microcontroller and a remote server is facilitated via Wi-Fi connectivity, enabling real-time physiological monitoring without physical tethers. An integrated Light Emitting Diode (LED) provides visual feedback regarding the system's operational status, including power availability and network connectivity, enhancing user awareness of the device's functionality. This design prioritizes stable power delivery and seamless data transmission This approach is inspired by previous designs for wearable IoT devices [41].

Software Environment

The ESP32 firmware was developed utilizing the Arduino Integrated Development Environment (IDE) and associated ESP32 core libraries. Data management and visualization were implemented using PHP, MySQL, and Apache, collectively known as the XAMPP stack. A locally hosted backend server facilitates the processing of incoming sensor data via a PHP script. This script is designed to receive Hypertext Transfer Protocol (HTTP) POST requests and subsequently insert the data into a structured MySQL database. Each entry within the database is timestamped, which enables time-series analysis and synchronization with the digital twin simulation. This approach allows for effective data tracking and integration with simulation models As shown in recent work using AI-based wearables [42], predictive monitoring enables personalized intervention.

Data Storage Schema

The database architecture incorporates a structured table for sensor data, meticulously designed to accommodate eight signal readings per sample. This includes six readings from Flex sensors and two from electrocardiogram (ECG) sensors. Each record is uniquely identified by an auto-incremented primary key and a corresponding timestamp, crucial for temporal analysis. The defined schema prioritizes data integrity and consistency, essential for the robust training and validation of artificial intelligence (AI) models utilized in subsequent analytical procedures. The underlying SQL structure facilitates efficient data storage and retrieval Pain intensity estimation from physiological signals has been explored in recent studies [43], Below is the SQL structure used:

```sql
CREATE TABLE sensor_data (
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  SF1 FLOAT NOT NULL,
  SF2 FLOAT NOT NULL,
  SF3 FLOAT NOT NULL,
  SF4 FLOAT NOT NULL,
  SF5 FLOAT NOT NULL,
  SF6 FLOAT NOT NULL,
  ecg1 FLOAT NOT NULL,
  ecg2 FLOAT NOT NULL,
  timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Listing 4.1: Creating the sensor$_d atatable$

4.2.2 Sensor Programming and Data Reading Using ESP32

The ESP32 microcontroller's firmware logic facilitates continuous acquisition, filtering, and transmission of sensor data to a designated database. This embedded system implementation streamlines data processing by integrating data acquisition, signal conditioning via filtering, and efficient data transfer protocols for seamless integration with a remote database.

Initialization and Pin Mapping

The Arduino script's initialization encompasses all analog and digital pins interfaced with the Flex and ECG sensors. The Flex sensors are individually mapped to the ADS1115's four channels, necessitating channel switching and appropriate gain configuration to optimize signal acquisition. Conversely, the ECG sensors are directly connected to analog pins and undergo amplification. This amplification is crucial to ensure the reliable detection of low-voltage cardiac signals, enhancing the accuracy and robustness of the physiological monitoring system.

```
// Example: Reading flex sensor 1 via ADS1115
float SF1 = ads.readADC_SingleEnded(0);  // Channel 0
```

Listing 4.2: Reading Flex Sensor Data

Signal Acquisition and Preprocessing

In accordance with established protocols for posture monitoring and pain detection, data acquisition was performed at a sampling rate of 1 Hz. The acquired analog signals underwent analog-to-digital conversion and were temporarily stored within the ESP32's memory module. To enhance data fidelity and minimize spurious noise artifacts, real-time implementation of fundamental noise reduction methodologies, specifically moving average smoothing, was applied to the digitized signals. This preprocessing step is crucial for ensuring the reliability and accuracy of subsequent data analysis.

HTTP Data Transmission Protocol

The ESP32 microcontroller aggregates collected data following each sampling cycle. Subsequently, it formulates an HTTP POST request, transmitting the data to a designated server via Wi-Fi connectivity. The server-side PHP backend undertakes the parsing, validation, and insertion of the received data into a MySQL database. To ensure reliable data transmission, an error-checking mechanism is implemented. In instances of transmission failure, the system initiates retry attempts to guarantee successful delivery. This robust approach facilitates accurate and consistent data storage Wireless Body Area Networks (WBANs) have been extensively studied for healthcare solutions [44].

Example snippet from the PHP server code:

```
$sql = "INSERT INTO sensor_data (SF1, SF2, SF3, SF4, SF5, SF6, ecg1,
    ecg2) VALUES (...)";
```

Listing 4.3: Inserting Sensor Data into Database

Backend Database Architecture

To store and manage the physiological and contextual data collected by the smart vest system, a relational database was designed and implemented using MySQL. As shown in Figure 4.19, the system's database is structured to manage patients, sensor data, digital twin records, and physiotherapy sessions.

Figure 4.19: Entity Relationship Diagram (ERD) of the System Database

The database schema is composed of multiple interrelated tables that capture the modular structure and complexity of the patient monitoring system. At the center of the schema is the `Patients` table, which stores essential patient identifiers and demographic information. This table establishes relationships with other key entities such as `Medical_Information`, Vital_Signs, and Contact_Information, allowing the system to provide a comprehensive view of each patient's health status.

The Smart_Vest table logs timestamped sensor data for each patient, including electrocardiogram readings (SECG1, SECG2) and six flex sensor values (SF1âSF6), in addition to a calculated pain score. These sensor records are linked to 3D scan data and digital twin representations through the 3D_Data and Digital_Twin tables, enabling synchronized updates of virtual patient models.

Diagnostic imaging is managed through the Scan_Data and Imaging_Requests tables, while rehabilitation activities are tracked via Physiotherapy_Sessions and Patient_Exercises. The inclusion of timestamps across all sensor readings and session logs ensures data traceability and supports longitudinal analysis of patient progress.

This normalized structure ensures scalability, data integrity, and efficient querying for both real-time monitoring and offline model training. Furthermore, the relational model supports linking physiological data to treatment plans and digital twin updates, enabling closed-loop feedback between the real and simulated patient environments.

System Synchronization and Timestamping

To maintain analytical rigor, each datum undergoes temporal indexing via the ESP32's internal clock, which is synchronized with Network Time Protocol (NTP) when network availability permits. This process establishes a temporal sequence essential for correlating data points with external visual feedback and digital twin simulation outputs. Such alignment facilitates a comprehensive and synchronized analysis of system behavior Posture monitoring has been explored using wearable systems such as smart vests [45].

Summary of Data Flow

The flow of sensor data follows this sequence, As shown in Figure 4.20, the data travels from sensors through several stages before being stored in the database.

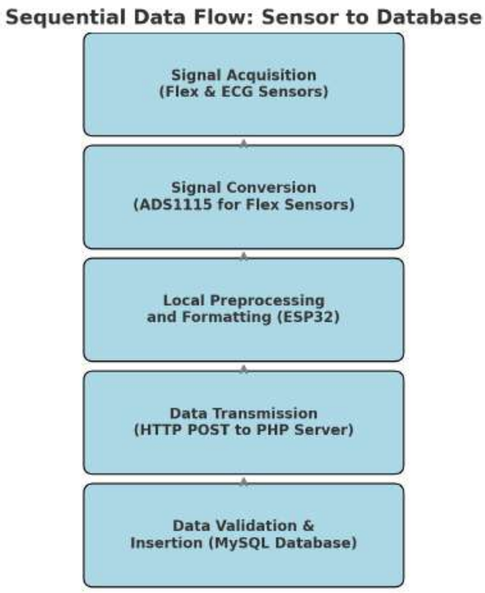


Figure 4.20: Sequential Data Flow from Sensor to Database

4.2.3 Data Processing and AI Model Preparation

This section outline the methodologies employed to analyze sensor data acquired from a smart vest, culminating in the development of an artificial intelligence model for pain level classification. A Convolutional Neural Network (CNN) was constructed and trained to interpret multimodal sensor signals, specifically electrocardiogram (ECG) and flex sensor data, to differentiate between high and low pain states in patients. The subsequent sections detail the model's architectural design, the training pipeline implemented, the optimization strategies utilized to enhance performance, and the evaluation metrics employed to assess the model's efficacy and robustness in pain level classification.

Model Architecture Design

A one-dimensional convolutional neural network (1D-CNN) architecture was employed to construct the artificial intelligence (AI) model, enabling it to process time-series data. The model's input comprises synchronized sensor readings from electrocardiogram (ECG) and flex sensors, collected across a temporal dimension. These signals undergo preprocessing and are structured into arrays to ensure compatibility with neural network input requirements. The convolutional neural network (CNN) architecture comprises several layers, facilitating the extraction of relevant features and patterns from the preprocessed sensor data. The layers are structured as shown in Figure 4.21, where the CNN architecture processes time-series sensor data for effective pain classification.

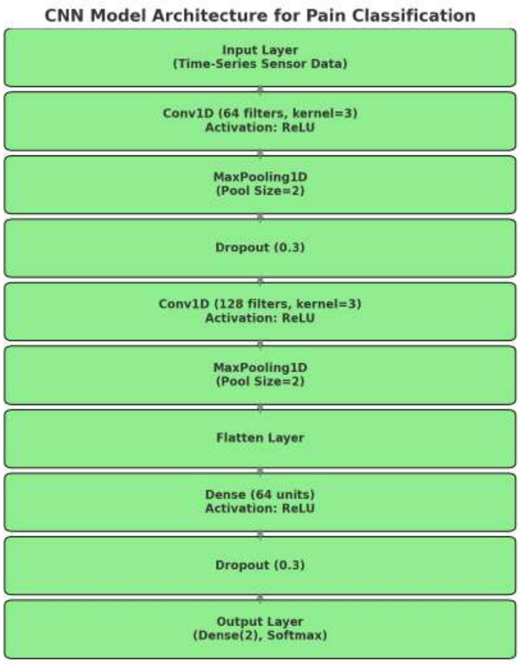


Figure 4.21: CNN Model Layers Architecture

1. **Input Layer:** Receives raw sensor data formatted as time-series inputs.

2. **First Convolutional Layer:** `Conv1D(64, kernel_size=3, activation='relu')` to extract low-level features from the raw signal data.

3. **First MaxPooling Layer:** `MaxPooling1D(pool_size=2)` to reduce dimensionality and retain key features.

4. **Dropout Layer:** `Dropout(0.3)` to minimize overfitting by randomly deactivating 30% of neurons during training.

5. **Second Convolutional Layer:** `Conv1D(128, kernel_size=3, activation='relu')` for capturing more complex signal patterns.

6. **Second MaxPooling Layer:** Additional `MaxPooling1D(pool_size=2)` to downsample the extracted features.

7. **Flatten Layer:** `Flatten()` to convert the pooled feature maps into a 1D vector.

8. **Dense Hidden Layer:** `Dense(64, activation='relu')` to further interpret the extracted features.

9. **Dropout Layer:** `Dropout(0.3)` applied again to regularize the model.

10. **Output Layer:** `Dense(2, activation='softmax')` for binary classification (low pain vs. high pain).

Dataset Preparation and Labeling

The sensor data acquired was meticulously structured into labeled datasets. Each data sample correlated to a distinct pain level, as determined through subject self-assessments and postural indicators. This comprehensive dataset was subsequently partitioned to facilitate rigorous analysis and model training. This allows for better assessment of the data that was provided. The dataset was split as follows: 80% for training and 20% for testing, as illustrated in Figure 4.22.

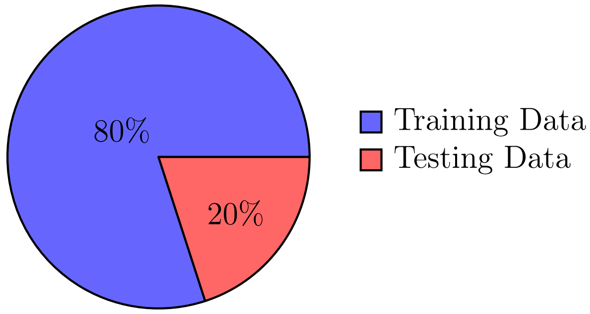


Figure 4.22: Dataset Split

To facilitate optimal model training and convergence, data normalization was implemented to standardize input ranges prior to model application.

Training Configuration

The model was compiled and trained using the following hyperparameters and configuration:

1. **Optimizer:** `Adam`, chosen for its adaptive learning capabilities.
2. **Loss Function:** `categorical_crossentropy`, suitable for multi-class classification tasks.
3. **Evaluation Metric:** `accuracy`, to monitor the modelâs predictive performance.
4. **Epochs:** 20 training iterations.
5. **Batch Size:** 32 samples per batch.

The training was executed using the Keras deep learning framework in Python:

```
history = model.fit(X_train, y_train, epochs=20, batch_size=32,
    validation_data=(X_test, y_test))
```

Listing 4.4: Training the CNN Model

Model Optimization and Hyperparameter Tuning

To optimize performance, multiple tuning strategies were implemented, demonstrating a commitment to enhancing system efficiency and effectiveness:

1. **Architecture Selection:** Several combinations of convolutional and dense layers were tested to identify the most effective structure.

2. **Cross-Validation:** K-fold cross-validation was used to assess the modelâs generalizability across different subsets of the data.

3. **Learning Rate Adjustment:** Various learning rates were tested to find the balance between fast convergence and model stability.

4. **Overfitting Control:** Dropout layers and batch normalization were used to reduce the risk of overfitting.

5. **Normalization:** Applied across all input channels to ensure signal uniformity and support gradient-based optimization.

Applicability

The Convolutional Neural Network (CNN) model exhibits promising capabilities in the accurate classification of pain levels derived from sensor data analysis. This suggests potential applications in supporting physical therapy practices through the creation of digital twin environments capable of simulating patient responses to diverse movements and therapeutic interventions. The integration of sensor-driven Artificial Intelligence (AI) within this framework facilitates improved clinical decision-making by providing a data-driven approach to understanding patient responses. This further allows for the development of safer, more informed rehabilitation plans that can be rigorously evaluated in a virtual setting prior to their physical implementation.

4.2.4 Model Training Using Real Data with Python and Keras

The subsequent section delineates the training methodology employed for the Convolutional Neural Network (CNN) model, utilizing empirical physiological data acquired via a wearable system. The implementation is executed in Python, capitalizing on prominent machine learning libraries such as TensorFlow, Keras, NumPy, pandas, and scikit-learn for data processing and model development.

Data Preprocessing and Label Encoding

A comprehensive dataset was constructed from sensor data obtained during prolonged periods of both optimal and suboptimal postures. This dataset, saved in a .csv format, incorporates average electrocardiogram (ECG) values and average flex sensor readings, correlated with pain level As shown in Figure 4.23, the dataset includes samples distributed across low, medium, and high pain levels.

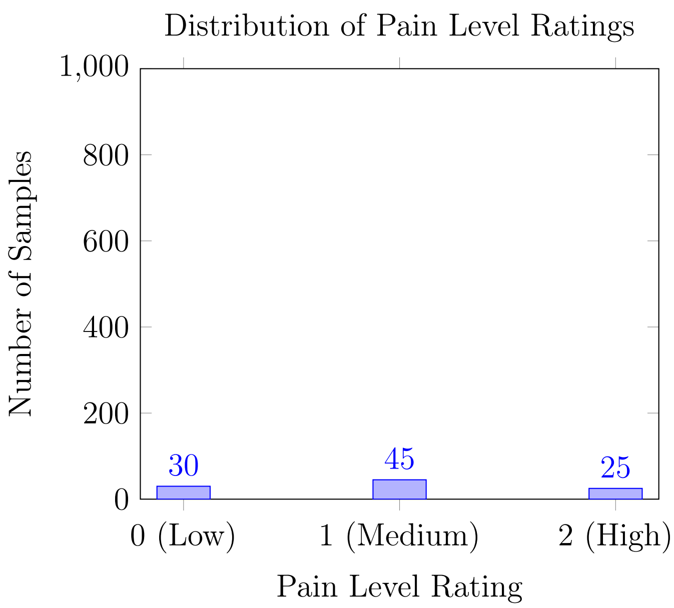


Figure 4.23: Bar Chart Showing Pain Level Ratings

Data pre-processing involved the removal of missing values and the application of StandardScaler for data normalization to define input features. Furthermore, classifications underwentSingle-encoding was performed using the `to_categorical()` function, facilitating subsequent multi-class classification analysis.

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_scaled = X_scaled.reshape(X_scaled.shape[0], X_scaled.shape[1], 1)
```

Listing 4.5: Standardizing and Reshaping Input Data

Model Architecture and Training Configuration

A Convolutional Neural Network (CNN) model was developed utilizing the Keras Sequential API. The architecture incorporates a one-dimensional convolutional layer, featuring 64 filters with a kernel size of 2, succeeded by a max-pooling layer for dimensionality reduction. Dropout regularization was applied to mitigate overfitting. The resulting flattened features were then fed into a densely connected hidden layer comprising 64 neurons and employing Rectified Linear Unit (ReLU) activation. Further dropout was implemented before the final softmax output layer, which consisted of three neurons representing the three pain levels under consideration.

```
model = Sequential([
    Conv1D(64, kernel_size=2, activation='relu', input_shape=(X_scaled.
    shape[1], 1)),
    MaxPooling1D(pool_size=1),
    Dropout(0.3),
    Flatten(),
    Dense(64, activation='relu'),
    Dropout(0.3),
    Dense(3, activation='softmax')
])
```

Listing 4.6: Sequential CNN Model Architecture

A model was constructed utilizing the Adam optimization algorithm and categorical cross-entropy loss function. The training regimen consisted of 20 epochs, employing a batch size of 16. Performance evaluation during training was conducted through the continuous monitoring of both accuracy metrics and validation loss values.

```
model.compile(optimizer='adam', loss='categorical_crossentropy',
    metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=20, batch_size=16,
    validation_data=(X_test, y_test))
```

Listing 4.7: Compiling and Training the CNN Model

Saving the Model and Scaler

Following the training phase, the model was saved in HDF5 format. To maintain preprocessing consistency during real-time inference, the scaler was preserved utilizing the joblib library. This approach ensures reliable and reproducible model performance in subsequent applications.

```
model.save("pain_prediction_cnn_model.h5")
joblib.dump(scaler, 'scaler.pkl')
```

Listing 4.8: Saving the Trained Model and Scaler

4.2.5 Integrating the Model with the Physical System

To facilitate real-time pain level prediction during patient interaction with the smart vest, the trained model was implemented through a lightweight RESTful API utilizing the Flask framework. This deployment enables efficient and accessible pain assessment.

Flask-Based API Deployment

The application programming interface (API) was developed utilizing the Python programming language and the Flask framework, further enhanced with the Flask-CORS extension to facilitate cross-origin communication with client-side applications. For operational efficiency, the API initializes by loading the pre-trained convolutional neural network (CNN) model and the associated scaler file (scaler.pkl). This ensures consistent data scaling and reliable inference throughout the application's lifecycle.

```
model = load_model('pain_prediction_cnn_model.h5')
scaler = joblib.load("scaler.pkl")
```

Listing 4.9: Loading the Saved CNN Model and Scaler

Inference and Prediction Endpoint

The system employs a POST endpoint that receives JSON payloads encompassing average electrocardiogram (ECG) and flex sensor data. Pre-processing steps involve reshaping and scaling these input values before they are submitted to a Convolutional Neural Network (CNN) for inference. Subsequently, a proprietary interpretation function decodes the softmax output to determine the corresponding pain level category.

```
input_array = np.array(input_data).reshape(1, -1)
scaled_input = scaler.transform(input_array)
scaled_input = scaled_input.reshape(1, 2, 1)
prediction = model.predict(scaled_input)
```

Listing 4.10: Preparing Input and Making Predictions

The predicted pain level is returned as a human-readable string: "Low", "Medium", or "High".

Language and Technology Stack

The complete programming stack for model training and integration includes:

1. **Python:** Primary language for data processing, model training, and backend deployment.
2. **TensorFlow/Keras:** Used for defining and training the CNN model.
3. **scikit-learn:** Utilized for data scaling and dataset splitting.
4. **Flask:** Lightweight web framework used to build the API for model deployment.
5. **joblib:** Used to serialize and save the trained scaler object.

The environment dependencies are managed via, ensuring that versions of all libraries are compatible and reproducible.

Real-Time System Interaction

The API facilitates real-time data transmission from sensors to clinicians via Wi-Fi from the ESP32 or a local client. This integration provides immediate feedback on a patient's pain status during exercises or rehabilitation. Furthermore, the system enables seamless synchronization with a digital twin, facilitating visualization and longitudinal tracking of patient data. This contributes to enhanced monitoring and data-driven clinical decision-making.

4.3 Testing and Data Analysis

4.3.1 Testing Scenario

Objective of the Testing Scenario

The main objective of this testing scenario was to examine the effectiveness of the smart wearable system in identifying physiological responses related to pain, specifically those resulting from incorrect sitting postures. By analyzing real-time sensor data, the goal was to determine whether the system could reliably detect and differentiate between pain-inducing and non-painful postural conditions.

Participant Selection Criteria

Two participants were carefully selected to represent two distinct physical conditions, with the aim of enabling an accurate comparison of physiological responses related to posture.

The first participant had a long-term habit of sitting in an incorrect posture. Due to repeated exposure to this poor posture, neuromuscular adaptation occurred, causing the participant to experience pain when attempting to sit properly. As a result, this individual was classified as the experimental subject, representing the pain-related condition associated with improper sitting.

The second participant, on the other hand, had no history of musculoskeletal issues and was able to maintain a correct sitting posture without experiencing any discomfort or pain. Therefore, this participant was classified as the control subject, representing a normal, pain-free condition that serves as a baseline for comparison.

Standardized Testing Environment

The experiment was conducted in a controlled laboratory environment to ensure data validity and minimize the influence of external variables. Environmental factors such as lighting and room temperature were carefully adjusted to comfortable and appropriate levels to avoid any discomfort or physiological disturbance.

To reduce potential interference with the electrical signals, participants were instructed to keep their mobile phones away and avoid interacting with others during the session. This helped minimize cognitive or emotional distractions that could affect the accuracy of readings from ECG and Flex sensors. Additional measures were taken to ensure that each participant was physically relaxed and not experiencing any psychological stress or discomfort, allowing for the collection of clear and consistent signals.

Each testing session lasted two hours per participant. During this period, the experimental subject maintained an incorrect sitting posture known to cause pain and discomfort, while the control subject maintained a correct posture without showing any signs of pain or distress.

This carefully designed setting enabled the collection of reliable and consistent physiological data under two contrasting conditions: one associated with pain, and one without. It also allowed for a direct and meaningful comparison of ECG and Flex sensor signals linked to each postural state.

4.3.2 Performing Movements and Monitoring System Response

The Role of Movement Variations in System Evaluation

This stage of the experiment did not rely solely on fixed sitting positions. Instead, the evaluation was expanded to include intentional transitional movements while seated.

These movements included gradual leaning forward, lateral bending, and alternating between correct and incorrect postures. The goal was to simulate common motor behaviors that typically occur in natural sitting environments, whether at home, at work, or in everyday settings.

These postural variations were introduced in a deliberate and structured manner to measure the systemâs sensitivity to subtle deviations. The experiment focused on testing the responsiveness of the Flex sensors in detecting slight spinal bends, as well as the ECG sensors in capturing physiological changes triggered by posture shifts, such as elevated heart rate or increased muscle tension.

The systemâs ability to process this data in real time was also assessed, with emphasis on its consistency in recognizing both sudden and gradual changes. This aspect of the testing provided a precise view into how well the system could perform under realistic conditions, where sitting is not always static or uniform, but includes natural and often unpredictable movement.

In general, this phase of the experiment highlighted the importance of the interaction between movement and physiological data. It demonstrated the smart systemâs accuracy in capturing and interpreting these changes as they occur, thus validating its effectiveness and real-world applicability in dynamic postural scenarios.

Sensor Performance During Movement

During the transitions between different sitting postures, as well as during natural minor seated movements, the sensors embedded in the smart vest remained continuously active. This uninterrupted operation enabled the accurate and consistent recording of physiological data, allowing real-time tracking of changes in the userâs posture and physical state.

The Flex sensors played a central role in this stage of the experiment. Positioned along the spine, they were responsible for measuring angular deviations caused by spinal bending. These measurements helped the system detect even subtle misalignments from the ideal posture, enabling precise tracking of the participant's physical performance.

In parallel, the ECG sensors continuously monitored the electrical activity of the heart throughout the movement transitions. These signals provided valuable insight into the internal physiological responses, such as elevated heart rate or neuromuscular tension, which may result from uncomfortable or physically demanding postures.

The integration of both mechanical and physiological signals enabled the system to deliver a more comprehensive analysis of the userâs condition. It also supported the evaluation of the systemâs responsiveness to real-life postural variations encountered during daily seated activities.

Real-Time System Monitoring

The ESP32 microcontroller within the system played a central role in managing data flow during the experiment. It was responsible for processing the incoming signals from the various sensors in real time and wirelessly transmitting the data to a connected monitoring platform. This real-time connectivity allowed for the continuous tracking of the systemâs performance throughout movement phases, and enabled immediate observation of any physiological changes as they occurred.

In addition to data transmission, the system included LED indicators, which were utilized to provide visual confirmation of device status. These indicators signaled successful operation and connectivity during the entire evaluation session, ensuring transparency and reliability in system performance monitoring.

4.3.3 Collected Data Properties (ECG, Flex Sensors)

Types of Physiological Data

The physiological data collected in this study were derived from two primary sensor types embedded within the smart vest: electrocardiogram (ECG) signals and Flex sensor readings.

ECG sensors provided high-resolution data on the heartâs electrical activity. These signals are highly sensitive to internal physiological changes, often influenced by physical strain, emotional stress, or discomfort, making them valuable indicators in pain-related scenarios.

Flex sensors, on the other hand, measured the angular displacement of the spine. These readings offered a real-time mechanical representation of the userâs posture, allowing for the detection of subtle deviations from proper sitting alignment.

By combining both sources, the system was able to capture a comprehensive picture of the userâs physiological state from both internal and external perspectives.

Data Sampling and Organization

All physiological signals collected from the ECG and Flex sensors were acquired at a fixed sampling rate of 1 Hz, equivalent to one data point per second. This frequency was carefully selected to ensure sufficient temporal resolution for tracking both gradual and sudden variations in physiological activity, particularly those associated with posture transitions or discomfort.

The 1 Hz rate offered a balanced trade-off between data volume and analytical precision, producing a dataset rich enough to capture relevant trends without becoming overwhelming in size. This balance was critical for maintaining the efficiency and reliability of subsequent data processing and modeling.

Following the data acquisition process, all collected signals were systematically organized into separate files corresponding to each participantâs sitting posture. Data associated with correct posture were stored independently from those recorded during incorrect posture, allowing for clear differentiation between the two behavioral states.

Each file contained sequential time-series entries from both ECG and Flex sensors, ensuring synchronized signal representation over time. This structured data format facilitated seamless integration into neural network-based analysis workflows, where it could be used directly for classification and behavior recognition without requiring further reformatting.

Initial Data Preprocessing

Before performing any data analysis, the collected datasets underwent a structured preprocessing phase to improve accuracy and ensure data reliability. This step involved: Removing missing values,Filtering out abnormal or inconsistent readings,Eliminating zero-value entries caused by sensor faults or disconnections.

Following the cleaning process, basic statistical measures such as mean values and trend analysis were applied to each signal. This helped in identifying patterns in the physiological responses across different postural conditions and prepared the data for further processing and model training.

4.3.4 Pain Classification Results: Algorithm Comparison

Algorithm Comparison

Several machine learning algorithms were evaluated in this study to classify pain levels using physiological signals from ECG and Flex sensors. Among these, Decision Tree (DT) and Random Forest (RF) achieved high performance metrics, such as accuracy, precision, and recall. However, both algorithms struggled to handle the inherent complexity and time-series nature of the physiological data. Decision Tree, for instance, is prone to overfitting in datasets with intricate nonlinear patterns, while Random Forest, though more robust, lacked the feature extraction capabilities necessary for optimal analysis of temporal signal data. Figures 4.24 and 4.25 illustrate the confusion matrices for the Decision Tree and Random Forest models, highlighting their classification performance.

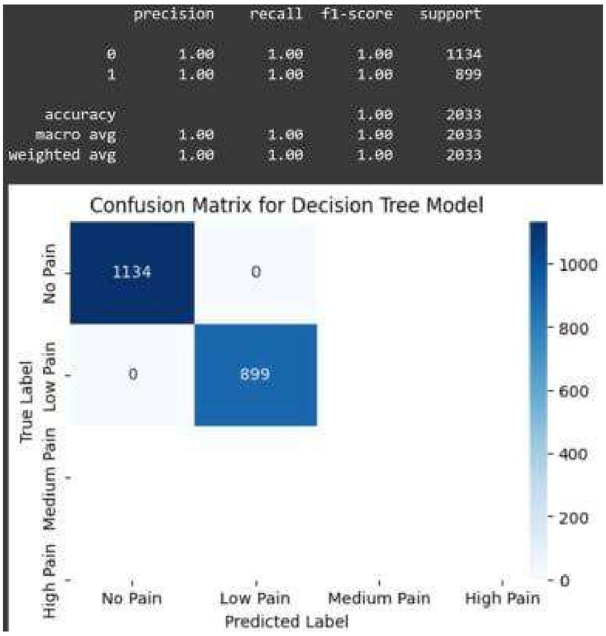
To further analyze the structure of physiological data, clustering methods such as K-means and Self-Organizing Maps (SOM) were applied. Figure 4.26 shows the K-means inertia evaluation curve, which was used to determine the optimal number of clusters in the dataset. Meanwhile, Figure 4.27 presents the SOM quantization error over epochs, demonstrating how the clustering model optimized data representation. These methods helped to uncover inherent patterns in the data distribution before applying classification models.

In contrast, the Convolutional Neural Network (CNN) demonstrated superior performance, not only in terms of accuracy but also in its ability to learn complex spatial and temporal relationships within the data. By leveraging automatic feature extraction, CNN effectively processed the 1D physiological signals, making it the most suitable choice for this application. The final model achieved an accuracy ranging between 91% and 92%, confirming its robustness and potential for real-time pain classification in wearable healthcare technologies.

Pain Classification

The models were applied to physiological data from ECG and Flex sensors. Traditional models showed poor generalization. In contrast, the CNN model achieved high performance with 92% training accuracy and 91% validation accuracy, with stable accuracy and loss curves (Figure 4.28).

Signal analysis revealed clear differences between low and high pain levels, highlighting the CNNâs ability to capture subtle patterns. The confusion matrix for the Decision Tree model Figure 4.24 and the Random Forest model Figure 4.25 also confirmed their reliability in classifying unseen cases, making them suitable for real-time clinical pain monitoring. A detailed discussion of pain classification and its results will be presented inA detailed discussion of pain classification and its results will be presented inA detailed discussion of pain classification and its results will be presented in Section 4.4 under the subsection 4.4.3.



	precision	recall	f1-score	support
0	1.00	1.00	1.00	1134
1	1.00	1.00	1.00	899
accuracy			1.00	2033
macro avg	1.00	1.00	1.00	2033
weighted avg	1.00	1.00	1.00	2033

Figure 4.24: Confusion Matrix for Decision Tree Model.

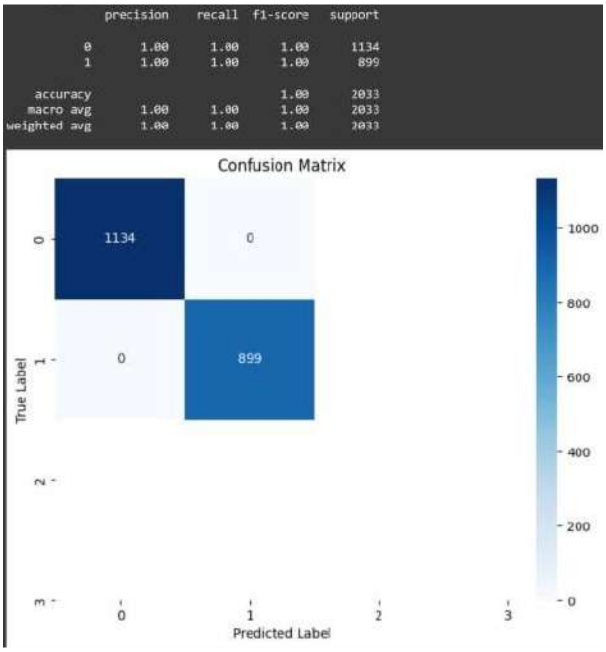


Figure 4.25: Confusion Matrix for Random Forest Model.

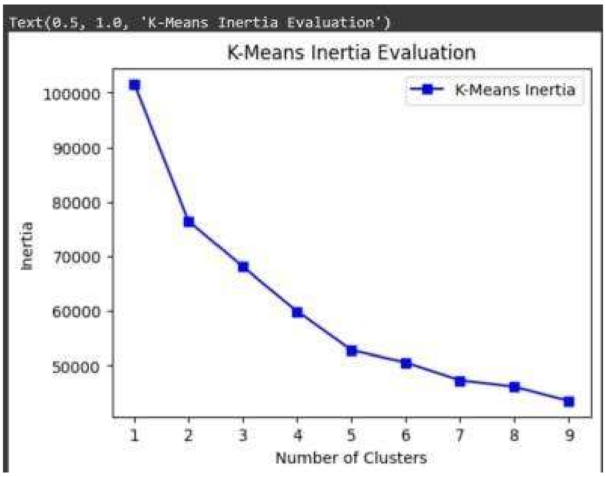


Figure 4.26: K-means Inertia Evaluation Curve.

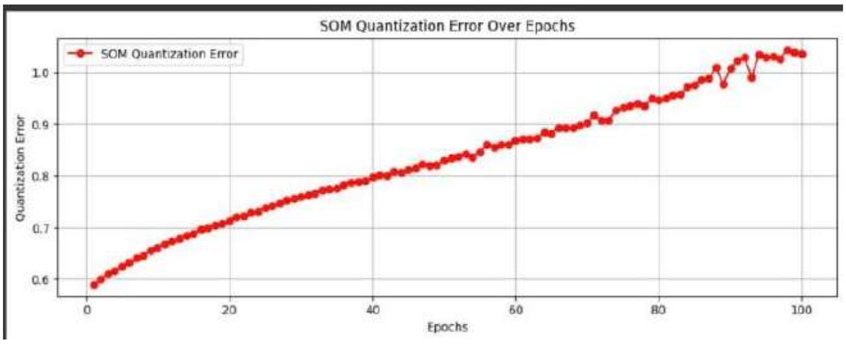


Figure 4.27: SOM Quantization Error Over Epochs.

4.4 Results Interpretation

4.4.1 Error Analysis and Model Interpretation

This section aims to provide a detailed analysis of the results obtained from the experiment or model. The data generated by the model is interpreted based on the criteria used to evaluate performance, with a focus on understanding the patterns or trends that emerge from these results. It also discusses how well these results align with the predefined objectives and evaluates the system's effectiveness in achieving the desired outcomes. This analysis also reflects the model's ability to handle real-world data and derive practical insights that can be useful in the future.

In this project, we relied on the Keras library to train a Convolutional Neural Network (CNN) model aimed at classifying a patientâs pain level using physiological data extracted from body-worn sensors, such as electrocardiogram (ECG) signals and flex sensors. Keras is one of the leading deep learning libraries, known for its ease of use and flexibility in designing multi-layer neural network models, especially when working with sensitive biomedical data.

To evaluate the model's performance, we used the Accuracy metric, calculated according to the following **Formula:**

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

The steps to calculate the accuracy using the Keras model were as follows:

1. **Merging real-world data:**
   - Two CSV files were used, each containing sensor readings from the same patient in two different physical postures:
     - Good posture → Assigned label `Pain_Level = 0`
     - Poor posture → Assigned label `Pain_Level = 2`
   - The two datasets were then merged into a single unified dataframe.
2. **Feature extraction:**
   - The average ECG signal was computed as `ECG_Avg` from the columns `ecg1` and `ecg2`.
   - The average flex sensor value was computed as `Flex_Avg` from columns `SF1 to SF6`.
   - These two aggregated features served as the primary input variables for the model.
3. **Scaling and splitting the data:**
   - The `Scikit-learn` library was used to split the dataset into training and testing sets with an 80% / 20% ratio.

- The input features were scaled using `StandardScaler` to improve learning efficiency and model convergence.

4. **Making predictions on the test set:**
   - After training the CNN model on the training data, predictions were made on the test set using:
     - `model.predict(X_test)`
   - The output probabilities were then converted into class labels using the `argmax` function.

5. **Calculating the number of correct predictions:**
   - The predicted class labels were compared with the true labels:
     - Correct predictions = 1871

6. **Total number of predictions:**
   - This corresponds to the number of test samples:
     - Total predictions = 2033

7. **Applying the accuracy formula:**
   - Accuracy $= \frac{1871}{2033} \times 100 \approx 92.03\%$

The CNN model demonstrated strong performance in predicting pain levels based on sensor data, achieving a final accuracy of 92.03

4.4.2 Comparison Between Expected and Actual Results

When comparing the expected results to the actual results of the Convolutional Neural Network (CNN) model in classifying pain levels, it can be observed that the model achieved a training accuracy of 92%. This means that it was able to accurately predict 92 out of every 100 cases during the training phase, reflecting a strong performance and exceptional ability to learn complex patterns from the available data. This percentage demonstrates the model's effectiveness in absorbing the information and physiological characteristics associated with different pain levels, enabling it to distinguish cases accurately. On the other hand, the validation accuracy was 91%, indicating that the model successfully classified 91 out of every 100 cases in the test set. This percentage suggests the model's ability to generalize the knowledge gained from the training data to new data, enhancing confidence in its efficiency and ability to perform in unfamiliar environments. Achieving such high accuracy indicates that the model is not only effective in learning from the dataset used but also in reliably applying that knowledge to data it has not encountered before.

4.4.3 Presenting Pain Classification Results

Convolutional Neural Networks (CNNs) are an effective tool for processing images and physiological signals. This report aims to analyze the performance of the CNN model in classifying pain levels based on physiological signal data.

The model achieved a training accuracy of 92% and a validation accuracy of 91% after 20 epochs, with stable accuracy curves for both training and validation. Additionally, there was a continuous decrease in the loss curve, indicating strong learning behavior without overfitting.

These results demonstrate the capability of CNNs to effectively learn from complex physiological data, making them suitable for real-time pain assessment applications. To understand the variation in physiological signals across pain levels, the average sensor readings and temporal patterns were analyzed. Electrocardiogram (ECG) signals and Flex signals showed measurable changes between low and high pain states, reflecting the model's ability to capture subtle patterns in the data. The confusion matrix confirmed the model's capability to classify unobserved cases during training, enhancing its reliability and generalization ability. These results support the suitability of the CNN model for clinical applications requiring precise pain monitoring, such as intensive care units and rehabilitation centers. The CNN model was trained using a dataset split into 80% for training and 20% for testing. The Adam optimizer was used to update the weights, while the Cross-Entropy function served as the loss criterion to measure the gap between predictions and labels. The model was trained for 20 epochs with a batch size of 32, allowing it to learn nonlinear patterns in the physiological signals.A summary of performance metrics for each epoch (including accuracy and loss) was tracked closely during training 4.1 During training, performance metrics, including accuracy, were monitored to provide insights into the model's progress. The model was tested on the test set to evaluate its ability to generalize predictions to unseen data. The CNN model demonstrates significant effectiveness in classifying pain levels based on physiological data, making it a powerful tool in clinical applications. The results highlight the importance of using convolutional neural networks to enhance pain monitoring and provide better healthcare [46].

Table 4.1: Different Environments Uses Different Algorithm

Epoch	Training Acc.	Validation Acc.	Training Loss	Validation Loss
1	0.9275	1.2000	0.2153	0.0020
2	0.9961	0.9995	0.0109	0.0020
3	0.9991	0.9999	0.0163	0.0007
4	0.9998	1.2000	0.0111	0.0051
5	0.9973	0.9999	0.0094	0.0013
6	0.9971	0.9995	0.0097	0.0013
7	0.9991	1.2000	0.0092	0.0007
8	0.9988	0.9999	0.0092	0.0005
9	0.9991	0.9999	0.0092	0.0007
10	0.9991	1.2000	0.0091	0.0002
11	0.9991	1.2000	0.0091	0.0002
12	0.9995	0.9999	0.0091	0.0002
13	0.9999	1.2000	0.0092	0.0005
14	0.9999	1.2000	0.0083	0.0003
15	0.9988	1.2000	0.0093	0.0005
16	0.9970	0.9997	0.0094	0.0018
17	0.9970	1.2000	0.0093	0.0003
18	0.9991	1.2000	0.0064	0.0003
19	0.9999	1.2000	0.0046	0.0001
20	0.9992	1.2000	0.0022	0.00005

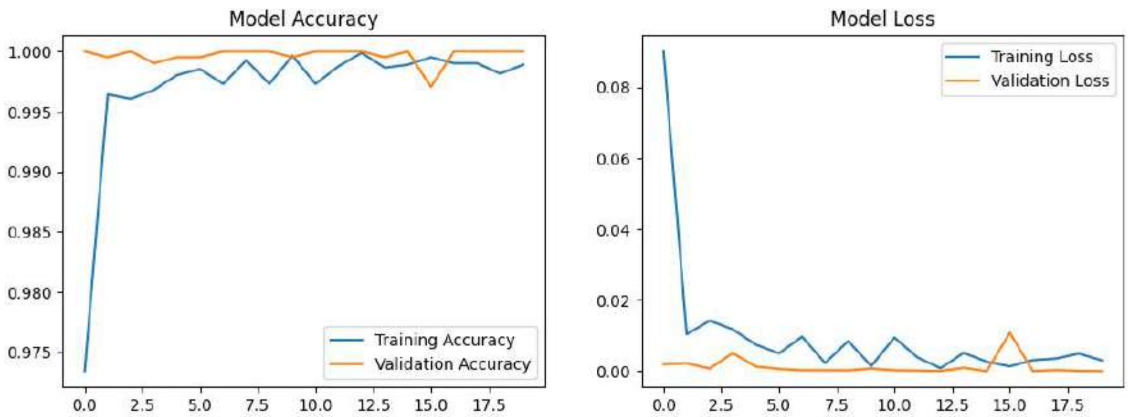


Figure 4.28: CNN Performance in Pain Classification

4.5 Challenges

1. Technical Challenges :

Linking Interfaces to the Database: We encountered difficulty ensuring that each interface interacted with the correct patient_id when sending or retrieving data.

In some cases, data failed to reach the server due to simple mistakes such as incorrect column names or invalid file paths.

Running the AI Model (CNN): The model required separating the front-end (interface) from the back-end (AI model) using Flask. The model wouldnât function unless the Flask server was running, which sometimes caused errors like: **Pain level** = **Error** when API requests were made without an active server.

Formatting and Sending Values to the API: In some instances, the sensor values received were null or invalid, which caused the API to reject the request.

2. Design Challenges:

Displaying the 3D patient model using Three.js required precise adjustment of position, rotation, and scale. Initially, the model appeared off-centered or out of view. To enhance interaction, OrbitControls were enabled for manual rotation. Notification display also faced issues where alerts either didnât show up, appeared in incorrect positions, or failed to link to buttons due to misconfigured IDs or CSS errors.

3. Runtime Challenges :

Runtime issues included interface errors caused by inactive Flask servers or invalid sensor inputs. Additionally, data format conflicts occurred during PHP integration where JSON format was required, but form-data was initially used, resulting in failed transmissions.

Chapter 5 | Conclusion and Future Works

5.1 Conclusion

In Conclusion, integrating digital twin technology in spinal rehabilitation represents a transformative leap in healthcare. By enabling physiotherapists to simulate and assess treatment plans virtually before application, this technology ensures precise and safe interventions. The proposed system highlights the potential for improved patient outcomes through personalized treatment, reduced recovery times, and minimized risks. As the healthcare sector continues to embrace innovation, the digital twin model exemplifies the future of patient-centric care.

5.2 Future Work

In capstone project, we provide a produce for the physiotherapist to try it on the digital twin before applying to the physical twin to ensure the properness of the exercises. In future work, the system should be able to estimate the actual recovery time, and help and support the physiotherapist to take decision.

References

[1] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," 2002.

[2] B. Agindra, H. B. Hidayati, L. Widhiyanto, and L. Herawati, "Profiles of patients with low back pain caused by disc herniation in the neurology outpatient clinic of dr. soetomo general academic hospital surabaya during the period of 2024," *Age*, vol. 21, no. 30, pp. 31–40.

[3] R. P. Di Fabio, G. Mackey, and J. B. Holte, "Physical therapy outcomes for patients receiving workers' compensation following treatment for herniated lumbar disc and mechanical low back pain syndrome," *Journal of Orthopaedic & Sports Physical Therapy*, vol. 23, no. 3, pp. 180–187, 1996.

[4] F. Tao *et al.*, "Digital twin-driven product design, manufacturing and service with big data," *The International Journal of Advanced Manufacturing Technology*, 2018.

[5] K. Bruynseels *et al.*, "Digital twins in healthcare: Ethical implications of an emerging engineering paradigm," *Frontiers in Genetics*, 2018.

[6] M. Batty, "Digital twins and smart cities," *Environment and Planning B: Urban Analytics and City Science*, 2018.

[7] A. Pokhrel, V. Katta, and R. Colomo-Palacios, "Digital twin for cybersecurity incident prediction: A multivocal literature review," in *IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW)*, (Seoul, Republic of Korea), 2020.

[8] E. PÃ¤rn, N. Ghadiminia, B. G. de Soto, and K. Oti-Sarpong, "A perfect storm: Digital twins, cybersecurity, and general contracting firms," *Developments in the Built Environment*, vol. 18, p. 100466, 2024.

[9] P. Jung, J. Murphy, and C. Zhang, "Digital twin-based testing for cyber-physical systems," *Journal of Systems Engineering*, vol. 12, pp. 123–134, Mar 2024.

[10] Smeets, "Digital twins for automotive development: Two-wheelers application," 2023.

[11] J. S. Kumar, S. B. Gosh, and T. Singh, "Blockchain in digital twin: Cybersecurity challenges and solutions," in *2023 IEEE International Conference on Emerging Technologies (ICET)*, (Abu Dhabi, United Arab Emirates), 2023.

[12] X. Gao, X. Zhang, and C. Li, "Neural network-based predictive modeling of chaotic systems with multi-step forecasting capabilities," *Scientific Reports*, vol. 13, no. 1, pp. 1–15, 2023.

[13] J. L. R. Nieuwoudt, P. J. Mahon, and A. G. White, "Real-time quantification of quantum coherence and entanglement with entangled photon pairs," *Scientific Reports*, vol. 13, no. 1, pp. 1–11, 2023.

[14] E. Bottani, R. Montanari, and M. Rinaldi, "The role of digital twin in industry 4.0 applications," *Proceedings of the International Conference on Advances in Manufacturing Technology*, 2017.

[15] R. Gupta, S. Khan, P. Roy, and S. Jain, "Emerging trends in machine learning-based cyber-physical systems: A review," *Journal of Manufacturing Systems*, vol. 57, pp. 1–12, 2024.

[16] A. Vallet, "Digital twins for healthcare systems," *Department of General Epidemiology, Foch Hospital, Suresnes, France*, 2020.

[17] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, and D. Devine, "Digital twin: Origin to future," *Applied System Innovation*, vol. 4, no. 2, p. 36, 2021.

[18] A. Hein, "Establishing a deep concept for self-defense surgery: A concept," 2020.

[19] A. S. Buonomo *et al.*, "Enhanced neuroimaging for dynamic biomedical conditions induced by radiation," *Scientific Reports*, 2020.

[20] A. Politi *et al.*, "Exploring the dynamic coherence of ecological networks," *Scientific Reports*, 2020.

[21] K. Bruinsels *et al.*, "Digital twins in healthcare: Ethical implications of an emerging engineering paradigm," *Frontiers in Genetics*, 2018.

[22] J. C. Doonan, A. M. Fahey, R. G. Bastos, and N. J. Turner, "Advances in precision medicine through novel targeted therapeutic approaches," *Expert Opinion on Drug Delivery*, 2024.

[23] L. Hein *et al.*, "Toward an integrated digital twin of shape and performance for lumbar spine analysis," *Digital Twin Journal*, vol. 1, no. 8, pp. 1–15, 2024.

[24] X. Gan *et al.*, "Ai-based framework for digital twins of vertebrae: Recongan," *Computational Neurology Magazine*, vol. 39, no. 2, pp. 101–115, 2024.

[25] M. Jones *et al.*, "Creating a digital twin of spinal surgery: A proof of concept," in *Proc. CVPRW 2024 Digital Twins in Healthcare*, pp. 123–134, 2024.

[26] U. of Sheffield, "About spinner: Enhancing spinal surgery with digital twins," 2024. [Accessed: Sep. 27, 2024].

[27] S. McBride *et al.*, "Digital twin models in centrifugal valve systems: A real-time performance tool," *Nature Scientific Reports*, vol. 11, no. 21, pp. 317–329, 2021.

[28] S. Smith *et al.*, "Digital twins in biomechanics: A path forward," *Journal of Biomedical Engineering*, vol. 10, no. 6, pp. 627–640, 2024.

[29] SparkFun Electronics, "Flex Sensor 2.2D̈atasheet." https://www.sparkfun.com/datasheets/Sensors/Flex/flex22.pdf, 2011. Accessed: 2024-11-22.

[30] Espressif Systems, "ESP32 Series Datasheet." https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, 2021. Accessed: 2024-11-22.

[31] Biosignalsplux by PLUX Wireless Biosignals, "ECG Sensor Datasheet." https://www.creact.co.jp/measure/bio/biosignalsplux/file/ECGSensor-Datasheet.pdf, 2020. Accessed: 2024-11-22.

[32] C. Ovalle Paulino and J. Huamani Correa, "Smart girdle to identify low back pain in patients with herniated disc through muscle pattern analysis," *Available at SSRN 4852586*.

[33] A. Kopelovich, M. Egwuenu, and M. Malik, "Lung cancer screening," *StatPearls*, 2023. Accessed: Nov. 24, 2024.

[34] C. Clinic, "Whole-body bone scan," 2024. Retrieved November 25, 2024.

[35] S.-H. Kim and M.-K. Park, "A study on efficient algorithm for 3d character animation," *Journal of the Korea Institute of Information and Communication*

Engineering, vol. 25, no. 10, pp. 1475–1481, 2021. Accessed: Nov. 22, 2024.

[36] F. S. U. Repository, "Advancements in animation technology for game development." https://repository.lib.fsu.edu/islandora/object/fsu:803218/datastream/PDF/view, 2020. Accessed: Nov. 22, 2024.

[37] W. Contributors, "Mixamo." https://en.wikipedia.org/wiki/Mixamo, n.d. Accessed: Nov. 22, 2024.

[38] J. H. Lee, "Unity: A powerful tool for 3d computer animation production," *Journal of the Korea Society of Computer and Information*, vol. 28, no. 3, pp. 1–10, 2023.

[39] B. Cutler *et al.*, "Advanced graphics final project: Fred the frog." https://www.cs.rpi.edu/~cutler/classes/advancedgraphics/S19/final_projects/fred.pdf, 2019. Accessed: Nov. 24, 2024.

[40] Scikit-learn Developers, "Supervised learning - Scikit-learn Documentation." https://scikit-learn.org/stable/supervised%20learning.html, 2024. Accessed: Feb. 11, 2025.

[41] N. A. S. Pangestu, D. I. Iryani, T. Febian, and M. R. Suryoputro, "Designing an iot-based vest for elderly people with dementia," in *Proceedings of the ICESH & Ergo-Camp 2023, ser. SHS Web of Conferences*, vol. 189, p. 01035, EDP Sciences, 2024. Accessed: Feb. 11, 2025.

[42] A. J. M. O. R. Khan, S. A. M. Islam, A. Sarkar, T. Islam, R. Paul, and M. S. Bari, "Real-time predictive health monitoring using ai-driven wearable sensors: Enhancing early detection and personalized interventions in chronic disease management," *International Journal for Multidisciplinary Research*, vol. 6, pp. 1–21, Oct. 2024. Accessed: Feb. 11, 2025.

[43] D. Albahdal, W. Aljebreen, and D. M. Ibrahim, "Painmeter: Automatic assessment of pain intensity levels from multiple physiological signals using machine learning," *IEEE Access*, vol. 12, pp. 48349–48365, 2024. Accessed: Feb. 11, 2025.

[44] G. A. Lopez, T. T. Y. Al-Naffouri, A. Alouini, and K. Y. Kabalan, "A survey on wireless body area networks for health-care applications," *Sensors*, vol. 20, no. 4, p. 969, 2020. Accessed: Feb. 11, 2025.

[45] H. He, W. Zhang, and D. Wu, "A smart vest system for lower back posture monitoring and risk assessment," *Sensors*, vol. 19, no. 20, p. 4503, 2019. Accessed: Feb. 11, 2025.

[46] K. N. Phan, N. K. Iyortsuun, S. Pant, H.-J. Yang, and S.-H. Kim, "Pain recognition with physiological signals using multi-level context information," *IEEE Access*, vol. 11, pp. 20114–20127, 2023.

Chapter 6 | Appendices A

Innovative Digital Twin Prototype for Spinal Rehabilitation

Demonstration Video

Publications

Patent

Patent Number: **SA 1020252406**