A Personalized Fitness Application for Injury Rehabilitation Using Artificial Neural Networks

Nikolaos Kalofonos, Christos Troussas, Akrivi Krouska, Phivos Mylonas, Cleo Sgouropoulou
Department of Informatics and Computer Engineering
University of West Attica
Egaleo, Greece
nkal01@hotmail.com, {ctrouss, akrouska, mylonasf, csgouro}@uniwa.gr

***Abstract*—This paper presents a personalized fitness mobile application with the use of Artificial Intelligence (AI), namely Artificial Neural Networks (ANN), which contributes to injury rehabilitation. It automatically generates customized workout plans on a weekly basis, taking into consideration user-supplied data on issues such as injury type, fitness level, body mass index, and available equipment. The system continuously updates the workout plans with the help of feedback provided by the user to achieve the best recovery process. Built with React Native for the frontend and Django for the backend, this application guarantees access anytime, anywhere, with smooth data treatment. The paper further reflects on the general implications of using AI in e-fitness, stresses intuitive user interfaces, and furthers possibilities such technologies could undergo.**

Keywords— AI in fitness; Artificial neural networks; Injury rehabilitation; Customized workout plans; Machine learning; Personalized fitness application

I. Introduction

The rise of e-fitness has completely revolutionized how people manage their physical health. In the early 2000s, fitness applications focused mainly on static training plans and basic monitoring of physical activities such as running or cycling. They initially gained popularity as basic tools that used smartphones to track physical activities such as walking and running. These early applications counted steps and calculated calories burned, offering users a simple way to track their daily activity levels. Fitness applications have evolved over the past ten years from simple activity trackers to complex systems that provide individualized coaching and training plans. Early e-fitness systems offered little insight into the unique demands of users and were mostly focused on static exercise routines. Fitness applications may now provide personalized suggestions based on a variety of user data points, including fitness level, age, and medical history, thanks to advances in artificial intelligence, particularly machine learning [1]. It has been observed that more than 60% of fitness center users used technology to perform their workouts. It has also been concluded that 72% of users considered that the use of exercise applications motivated them towards the achievement of their daily physical practice goals. In this sense, there is early evidence about applications that improve healthy lifestyle habits are well received by users [2].

In today's fitness technology, personalization has become a critical component. Conventional training platforms that are designed to fit all users are frequently ineffective because they do not take into consideration their individual goals and physical characteristics. Custom workouts may now be created by AI-based systems using data-driven methodologies, which is a big leap above previous fitness applications that only tracked basic variables. In this sense, artificial neural networks, a branch of machine learning, are especially potent since they can handle massive amounts of customized data and dynamically adjust to the profiles of individual users [3].

There are several real-world examples of AI-driven fitness applications. Workout Whiz utilizes artificial intelligence in order to offer individualized coaching and monitoring, making sure users complete workouts safely and correctly [4]. Another application with the same principles is trAIner, which also features automated coaching solutions by tracking exercise form, correcting it if it is wrong [5]. Verum Fitness uses the camera from a smart phone to record a user performing an exercise and skeletonizes them, extracts angles from specific joints, and feeds this data into a Fuzzy Inference System, to classify exercise performance [6]. Furthermore, research by LAZIER showed that the provision of online video instruction improves users' comprehension of how to complete assigned activities correctly [7].

According to additional research, the combination of wearables and other Internet of Things devices with artificial intelligence improves data collecting and accuracy, making it possible to personalize workouts even more precisely. These developments have made it possible for AI-powered fitness devices to provide diet plans, mental wellness support, and injury rehabilitation in addition to traditional physical training [8-9]. The rise of these applications represents a major change in the way people see fitness. Regardless of one's athletic background, these developments make leading a healthy and active lifestyle more accessible.

This paper aims to explore the development and implementation of a personalized fitness application that utilizes artificial intelligence, specifically an artificial neural network, to generate weekly customized rehabilitation plans for users recovering from certain injuries. By analyzing user data, such as injury type, body mass index (BMI), and equipment availability, the artificial neural network creates customized workout programs that adapt dynamically to direct user feedback. This research emphasizes the revolutionary role AI plays in injury recovery within the expanding e-fitness ecosystem through architectural analysis and case studies. Therefore, this project contributes to the increasing number of applications that aim to meaningfully incorporate AI into fitness and health. It offers a realistic and user-centric tool to improve rehabilitation outcomes for specific injuries.

The three main injuries this application helps rehabilitate are anterior cruciate ligament (ACL), meniscus, and achilles tendon damage. Surgical intervention is frequently necessary in cases of serious injuries, such as tendon tears, especially in younger individuals. Rehabilitation after surgery include a period of immobility and professional-supervised physical therapy. When the user is prepared for mild physical exercise, the application can be helpful; nevertheless, it is not meant to be used during the initial phase of rehabilitation. It offers users individualized exercises based on their injury type and fitness level to help them rebuild strength, mobility, and functionality, but it should only be used under medical advice.

II. Overview of the Application

The application created for this project will provide customized training plans to help users recover from injuries. Leveraging artificial intelligence capabilities, the application adapts to user inputs such as injury type, fitness level, and available equipment. At its core, the application operates as a smart fitness assistant that leverages an artificial neural network to interpret user-specific data and continuously optimize the training regimens through recurring user feedback. The main objective is to provide a personalized rehabilitation experience that makes the healing process effective and easy to access.

The fitness exercises included in the application, which concentrate on the three specific injuries mentioned in the introduction are sourced from medical research, physical therapy guidelines, and fitness forums. These exercises are categorized based on key factors such as injury type, fitness level, age, BMI, and required equipment. All this data is organized in a structured Excel sheet with direct connection to the application's database, serving as the central exercise dataset for all of its functions.

Every injury has a unique recovery schedule. Rehabilitation for ACL injuries usually consists of balancing exercises and strengthening of the hamstrings and quadriceps. Meniscus tears also concentrate on strengthening of the hamstrings and quadriceps but also on the flexibility and stability of the knee, frequently using low-impact exercises. Achilles Tendon injuries emphasize calf muscle strengthening and eccentric loading exercises. The exercises are classified according to the type of injury, the required fitness level depending on the recovery progress and the athletic ability, age, body mass index and equipment required.

This application is developed with Django for the backend, which controls user data and ANN connectivity, and React Native for the frontend, which ensures cross-platform compatibility on both Android and iOS devices. Scalability and flexible administration of both user data and the machine learning model are made possible by this separation of responsibilities. The user friendly interface is found in the application's intuitive layout which makes it easier to use, especially for people who aren't familiar with handling complex apps. The placement of the functions is organized in an easy to grasp manner, which enables smooth navigation throughout its interface. The system dynamically adjusts workout routines based on direct user feedback, allowing users to express their opinions on the plan's effectiveness. Interactions and updates are securely stored in the backend by modifying the existing user profiles, while the ANN processes this data to optimize exercise recommendations based on individual progress.

The client-server design of the fitness application ensures smooth communication between the frontend and backend components. Because of this design, the system can process user data efficiently in order to provide the workout plans. React Native, a popular cross-platform framework, is used in the development of the application's frontend, enabling seamless operation on both iOS and Android smartphones. React Native is well-known for its capability to provide native-like performance with only one codebase, which makes it perfect for applications that need to be scalable and easily maintainable.

The user interface is made to be simple to use, especially for users who may not be tech-savvy. Smooth screen transitions are ensured by the use of drawer and stack navigators, which facilitates user management over data within the application. Users can easily navigate through their personalized workout plans from the main screen of the application (as shown in Fig. 1). Exercises within each day's workout plan are provided with a thorough description that includes information on repetitions, target muscle groups, and any equipment needed. Users can view instructional visuals to verify appropriate form and mark workouts as completed. This clear layout ensures accessibility and keeps the focus on recovery and performance.

React Native's integration with JavaScript and native components allows real-time interactions, ensuring that users can quickly generate workout plans and receive immediate feedback on their performance. A form is used to enter data, such as the user's fitness level, preferred workouts, and injuries, and API calls are used to submit the data to the backend. Data integrity is ensured by the frontend, which also effectively handles exceptions and validation prior to submission. Other functions include:

- An Exercise Library which allows users to browse all available exercises in the application, filtering them by categories like strength or flexibility, as well as by specific muscle groups. Each exercise is accompanied by a description, repetitions, and required equipment.
- The User Profile page, where personal information such is stored, allowing easy changes to some of these in order to update future workout plans.
- Account Settings, where users can manage their login credentials, change their username or password, and delete their account if necessary.

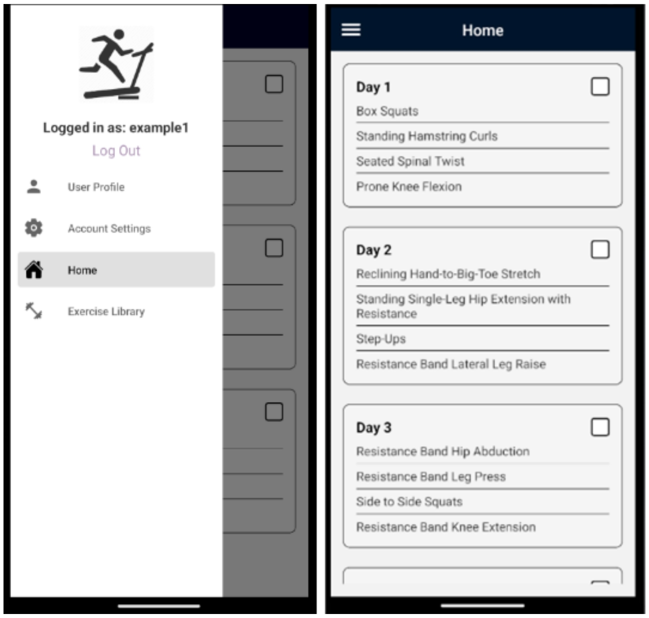

Fig. 1. Example of the main app interface.

Django, a Python-based framework that controls communication between the user interface, the databases and the ANN, is used in the construction of the application's backend. It handles user data, exercise information, user workout plans, and feedback using Django's Object-Relational Mapping, safely storing this data in a relational database, as in this case is SQLite. The admin interface lets developers directly handle data through an intuitive web interface, simplifying activities like viewing and editing records. The Django REST Framework offers APIs for communication between the frontend and backend. The communication flow of the system goes as follows: Users input their data via the React Native frontend, and this data is sent via Axios (HTTP client) to the Django backend through an API call. The Django backend processes the data, stores the information extracted in the database, and feeds it to the ANN for workout plan generation. The ANN evaluates the user's fitness data and generates personalized exercise recommendations grouped by days on a weekly plan. Finally, the backend sends the personalized workout plan back to the frontend, where it is displayed on the user interface for the user to interact with.

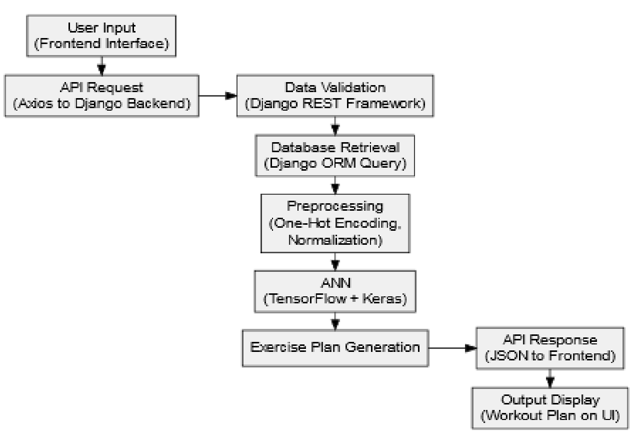

Fig. 2. Diagram of the data flow from user input to ANN output.

The app uses a feedback loop that allows users to directly influence the recommendations provided by the ANN. After marking all of the suggested workouts as completed, users can report their experience in order to proceed to next week's updated plan. The feedback choices include whether the workouts felt easy, doable, too difficult, or caused discomfort. This feedback is processed by the system, which uses it to adjust future workout plans. The most direct adjustments include altering the repetitions of each exercise, while repeated feedback of the same nature may upgrade or downgrade the overall fitness level of the user. For example, if a user reports discomfort, the ANN regenerates the plan with adjusted user information, replacing the exercises in subsequent sessions with lower difficulty ones, or if the user reports that they felt comfortable doing all of the exercises in subsequent feedback loops, the application suggest exercises of higher difficulty. Over time, this dynamic feedback loop helps the ANN refine its recommendations, in order to ensure that the workout plans evolve alongside the user's recovery progress.

III. Overview of the Implemented Artificial Neural Network

As previously mentioned, at the center of the system lies an artificial neural network (Fig. 3) that controls the personalization functions of the application. ANNs are computational models inspired by the way the human brain processes information [10-14]. They are trained in a way that makes them especially good at identifying patterns and changing with time, thus making them ideal for tasks such as classifications and predictions. Generally, they are composed of parts called layers, with each layer consisting of a number of neurons, where the three main are the input layer, the hidden layers and the output layer. The input layer receives the necessary user data, the hidden layers perform complex computations to determine optimal predictions and the output layer provides the results [15]. For each neuron in layer l, the input $x^{(l-1)}$ from the previous layer is combined with the weights $W^{(l)}$ and bias $b^{(l)}$:

$$z^{(l)} = W^{(l)} x^{(l-1)} + b^{(l)} \tag{1}$$

The application's ANN analyzes a number of crucial input variables, including the user's age, body mass index, fitness level, type of injury, and available equipment. Having been trained on a large dataset of exercises combined with numerous examples of users whose information matches several different combinations of the variables mentioned above. By modifying the network's internal parameters (weights) through a process known as backpropagation, the network gains the ability to correlate particular exercises with particular user profiles during the training phase.

Backpropagation is a supervised learning algorithm that optimizes the network's predictions by adjusting the weights of connections between neurons. This is done by calculating the gradient of the loss function (which measures the error in predictions) and propagating this gradient backward through the network. The weights are updated iteratively, minimizing the error over time. This process continues for several epochs, each consisting of a forward and backward pass through the network. During each pass, the learning rate controls how much the weights are adjusted, ensuring that the model converges to an optimal solution without overshooting. The network uses gradient descent to update its parameters:

$$W^{(l)} = W^{(l)} - \eta \frac{\partial L}{\partial W^{(l)}} \quad (2)$$

where η is the learning rate, and $\frac{\partial L}{\partial W^{(l)}}$ is the gradient of the loss function L with respect to the weights. The error is measured using the binary cross-entropy loss function:

$$L(y, \acute{y}) = -\frac{1}{n}\sum_{i-1}^{n} \left[y_i \log(\acute{y}_i) + (1-y_i)\log(1-\acute{y}_i)\right] \quad (3)$$

where y is the actual label, $\acute{y}$ is the predict probability, and n is the number of samples.

This allows the network to gradually improve its predictions. By using this technique, the ANN's accuracy can be increased while prediction mistakes are reduced. Thus, the ANN can receive the current user's personal information input, predict the optimal exercise recommendations and form the personalized weekly plan.

The model was designed with the help of TensorFlow, a machine learning open-source library, utilizing Keras, a high-level API that simplifies the process of training. Before the building and the training of the network, the preprocessing of the input data is necessary. That is achieved by separating them between Categorical and Numerical features. The categorical features are converted to binary vectors through One-Hot Encoding and the numerical features are scaled with StandardScaler. To evaluate the performance of the model and avoid overfitting, the dataset was divided into a training set and a test set. The training set was used to train the neural network, while the test set was held back to evaluate how well the model generalizes to new data.

After the preprocessing, three hidden layers are formed**,** and each layer uses the ReLU (Rectified Linear Unit) activation function, which helps the model handle complex relationships, deeming it a fully connected feedforward neural network. The input layer receives the processed features, which have been converted into numerical form through the preprocessing steps. There are three hidden layers, each with a decreasing number of neurons (128, 64 and 32). The ReLU activation function is applied to each hidden layer to introduce nonlinearity into the model. After each hidden layer, dropout layers are used to avoid overfitting. It randomly deactivates a fraction of neurons (30% in this case) during each training iteration, making the model more compact. The final output layer uses a sigmoid activation function, which outputs a probability between 0 and 1, indicating whether a particular exercise is suitable for the user. In this case the ReLU activation function is applied:

$$f(z) = max(0, z) \quad (4)$$

To make sure the model generalizes well to new data, cross-validation approaches were used in addition to dropout layers. The dataset was divided into training and validation sets so that the network's performance was monitored across various folds. By avoiding overfitting and fine-tuning hyperparameters, this method made sure that the model's correctness extended beyond the training set.

Once the architecture was defined, the model was compiled and trained using the Adam optimizer, which is known for its efficient and adaptive learning rates. Adam is an iterative optimization algorithm used to minimize the loss function during training. The binary cross-entropy loss function was used, as the solution is a binary classification problem (whether an exercise is suitable or not). The model was trained for 50 epochs, i.e. the neural network "went through" the entire dataset 50 times. This helped the model to gradually learn and improve its ability to predict appropriate exercises. After each epoch, the performance of the model was evaluated on the test set. This validation helped monitor overfitting, ensuring that the model would generalize well to new users. At the end of the training process, the model was evaluated on the test set to ensure that it did not overfit the training data and that it could be generalized to new users. The accuracy achieved in the test set determines how reliable the model is at recommending exercises.

The ReLU activation function was chosen for its ability to introduce non-linearity while being computationally efficient, allowing the network to handle complex relationships without encountering the vanishing gradient problem common in other activation functions. The Adam optimizer was chosen because it can adjust learning rates for any parameter, resulting in faster and more accurate convergence, while dropout was employed to assist the model generalize more effectively by arbitrarily disregarding some neurons during training.

After the ANN is trained, the generation of personalized workout plans follows. The system processes the exercise dataset in conjunction with user data that is retrieved from the database. Every exercise is given a suitability score by the model, which chooses the ones that are over a cutoff point (0.5) to be included. The software arranges workouts into daily sessions according to the user's weekly preferences.

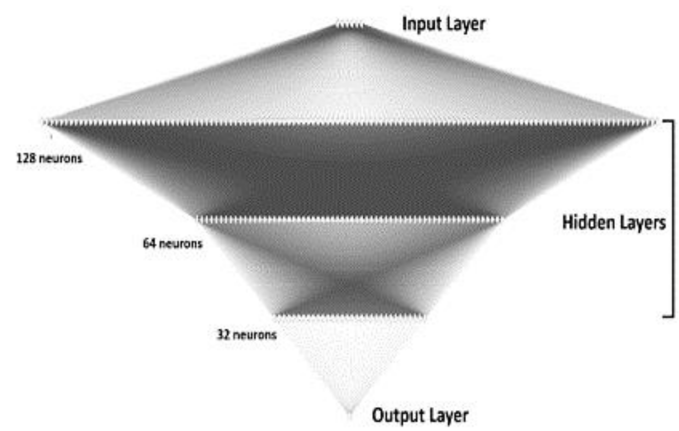

Fig. 3. Graphical representation of the ANN.

IV. Examples of Operation

The personalization capabilities of the application allow it to adjust based on the specific needs of each user, making it versatile for a variety of rehabilitation scenarios. Below are three examples that demonstrate how the application customizes workout plans for different use cases. The ANN will filter the exercises in terms of the suitability based on each user's specific input information, by using the pattern recognition knowledge it gained during the training phase.

User 1: An older user with an Achilles Tendon injury who has no equipment available

- Input: Male, Age 70, Height 1.82m, Weight 86kg, Beginner, Achilles Injury, 3 workouts per week, No available equipment

A plan is created for User 1, an elderly person without access to equipment who is healing from an Achilles tendon injury. It concentrates on fundamental leg stability, flexibility, and strengthening activities. The approach chooses exercises that can be performed without the use of equipment, such as calf raises, stretches, and toe curls, because Achilles injuries frequently necessitate a progressive reconditioning of the calf muscles and lower leg. The objective is to minimize the risk of overexertion while gradually strengthening the Achilles tendon area and increasing mobility. The software prioritizes recuperation over intensity, making ensuring the workouts are low-impact and suitable for older users, given the user's advanced age and novice fitness level (as shown in Fig. 4).

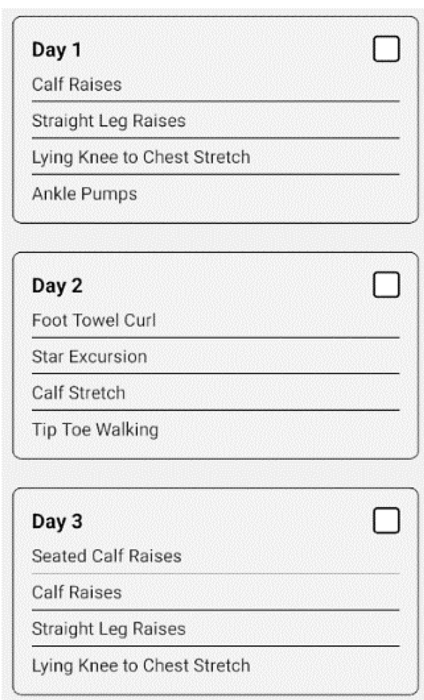

Fig. 4. Personalized plan for User 1.

User 2: Middle-Aged User with Old Meniscus Injury and Resistance Bands

- Input: Female, Age 51, Height 1.68m, Weight 63kg, Intermediate, Meniscus Injury, 4 workouts per week, Resistance Bands.

For User 2, who has a previous meniscus injury and access to resistance bands, a workout plan is generated. The plan focuses on knee support and strengthening the lower body. Strong quadriceps and glutes are needed to stabilize the knee joint and stop further damage in cases of meniscus problems. To strengthen specific muscular areas, the application suggests activities including squats, hip thrusts, and resistance band exercises. Exercises for stability and balance are also included in the plan to enhance total knee functionality. The exercises are more difficult than the ones of the previous example, as the user is at an intermediate level of fitness, and they include resistance bands for extra intensity. However, they are chosen carefully to prevent the overtiring of the knee joint (as shown in Fig. 5).

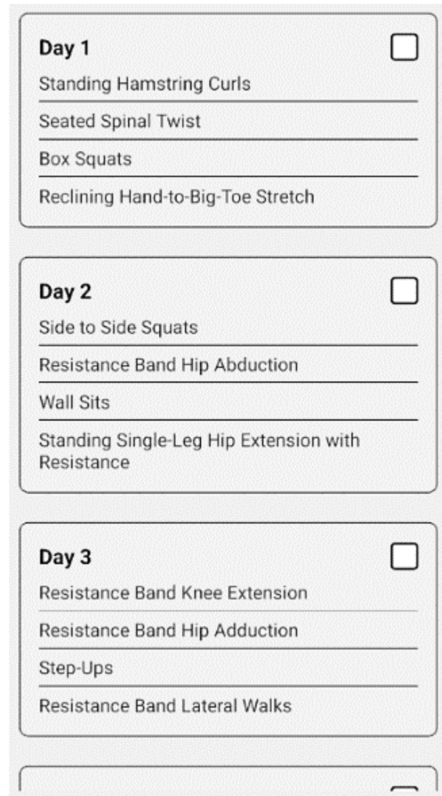

Fig. 5. Personalized plan for User 2.

User 3: Young Athlete in Late Stages of ACL Rehabilitation with Gym Equipment

- Input: Male, Age 24, Height 1.76m, Weight 70kg, Advanced, ACL Injury, 5 workouts per week, Gym Subscription.

For User 3, a young athlete nearing the end of his recovery from an ACL injury, the application generates a higher-level, high-intensity workout that includes using gym equipment. The goal is to strengthen and increase explosiveness in the legs, especially the quads and hamstrings, which are essential for knee stabilization. It is advised to perform exercises like leg presses, various lunges, and squats to build strength and improve knee flexibility and stability. Given the user's advanced ability level, the application includes more explosive and dynamic exercises—like plyometrics and box jumps—to assist the user regain full athletic function and agility. The method adjusts the level of difficulty and intensity to the user's ability while guaranteeing safe advancement through the latter phases of the rehabilitation (as shown in Fig. 6).

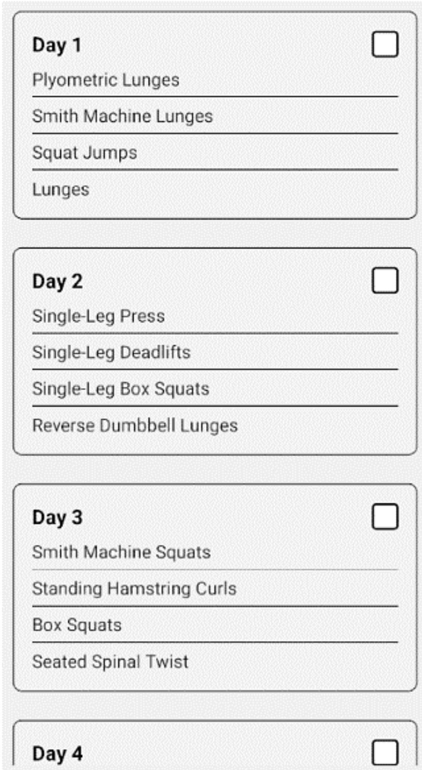

Fig. 6. Personalized plan for User 3.

In summary, the three rehabilitation use cases presented above highlight the application’s flexibility in modifying exercise regimens based on data given by the user. Individualized programs are made to meet the needs of every user, from young athletes in the later stages of recovery to elderly users with limited equipment.

V. CONCLUSIONS

The presented fitness application is a step of development in the field of AI-based personalization of training routines designed specifically for the goal of injury rehabilitation. It offers individualized training programs based on the user’s profile, supporting both general fitness objectives and the particular requirements of patients. Users can safely return to physical activity after recent injuries or maintain fitness levels despite chronic problems while minimizing the risk of re-injury. Its intuitive interface, robust neural network assistance, and real-time modifications based on user feedback enable users to take control of their fitness recovery journey.

By providing accessible and reasonably priced solutions to a much larger audience, this application has the potential to completely transform the field of injury rehabilitation, in terms of long-term impact. It has the potential to reduce the reliance on face-to-face physiotherapy sessions by providing structured, science-based exercise programs that are adjusted in real-time based on the user's progress. Rather than this though, the development of this technology could also act as a very useful adjunct at the disposal of specialist professionals helping their patients. Ultimately, this might result in quicker, more efficient recoveries, which would save healthcare expenses.

The application has significant growth potential, with planned improvements to expand functionality. One potential addition is the integration of medical advice into the rehabilitation process, offering guidance from physiotherapists alongside the exercise recommendations. The application could also evolve into a useful tool for clinics, supporting healthcare professionals in guiding patients through injury recovery. Another important addition could be a manual plan customization feature which would allow users to adjust their workout plans by selecting specific exercises from the library, providing another aspect of personalization.

Future expansions could include the addition of more exercises to support the recovery of plenty other injuries, enhancing the application’s versatility. The creation of more general fitness plans could also be supported, such as for muscle building or weight loss, potentially paired with customized nutrition plans. Exercise suggestions may be improved by integrating more complex machine learning models, such as deep learning or reinforcement learning models. Furthermore, the application would be able to provide even more accurate workout plan modifications or real-time exercise form correction if it integrated Internet of Things sensors, like a smartwatch or a camera, to record real-time data, such as movement, heart rate, and muscle activity.

REFERENCES

[1] A. Yadav, M. Choudhary, P. Gautam, K. Sankhe and A. Sirsat, "Workout Prediction Using Machine Learning," *2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, Salem, India, 2023, pp. 123-126, doi: 10.1109/ICPCSN58827.2023.00026.

[2] Valcarce-Torrente M, Javaloyes V, Gallardo L, García-Fernández J, Planas-Anzano A. Influence of Fitness Apps on Sports Habits, Satisfaction, and Intentions to Stay in Fitness Center Users: An Experimental Study. Int J Environ Res Public Health. 2021 Oct 2;18(19):10393. doi: 10.3390/ijerph181910393. PMID: 34639692; PMCID: PMC8507994.

[3] A. Flores, B. Hall, L. Carter, M. Lanum, R. Narahari and G. Goodman, "Verum Fitness: An AI Powered Mobile Fitness Safety and Improvement Application," *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, Washington, DC, USA, 2021, pp. 980-984, doi: 10.1109/ICTAI52525.2021.00156.

[4] A. Jadhav, S. Unkule, D. Hadsul, Y. Chiddarwar and P. Padhy, "Workout Whiz – Your Personalized AI PAL," *2023 6th International Conference on Advances in Science and Technology (ICAST)*, Mumbai, India, 2023, pp. 490-493, doi: 10.1109/ICAST59062.2023.10455053.

[5] V. Singh, A. Patade, G. Pawar and D. Hadsul, "trAIner - An AI Fitness Coach Solution," *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, Mumbai, India, 2022, pp. 1-4, doi: 10.1109/I2CT54291.2022.9824511.

[6] A. Flores, B. Hall, L. Carter, M. Lanum, R. Narahari and G. Goodman, "Verum Fitness: An AI Powered Mobile Fitness Safety and Improvement Application," *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, Washington, DC, USA, 2021, pp. 980-984, doi: 10.1109/ICTAI52525.2021.00156.

[7] R. Bi, D. Gao, X. Zeng and Q. Zhu, "LAZIER: A Virtual Fitness Coach Based on AI Technology," *2022 IEEE 5th International Conference on*

Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 2022, pp. 207-212, doi: 10.1109/ICISCAE55891.2022.9927664.

[8] F. Alshehri and G. Muhammad, "A Comprehensive Survey of the Internet of Things (IoT) and AI-Based Smart Healthcare," in *IEEE Access*, vol. 9, pp. 3660-3678, 2021, doi: 10.1109/ACCESS.2020.3047960.

[9] T. Galijašević, M. Škarić, E. Podolski, F. Mustač, M. Matovinović and D. Marčinko, "New Breakthroughs in AI Chatbots and Their Potential in Mental Health Services," *2024 9th International Conference on Smart and Sustainable Technologies (SpliTech)*, Bol and Split, Croatia, 2024, pp. 01-04, doi: 10.23919/SpliTech61897.2024.10612516.

[10] C. Troussas, F. Giannakas, C. Sgouropoulou, and I. Voyiatzis, "Collaborative activities recommendation based on students' collaborative learning styles using ANN and WSM," *Interactive Learning Environments*, vol. 31, no. 1, pp. 54–67, 2020. doi: 10.1080/10494820.2020.1761835.

[11] C. Troussas, A. Krouska, and M. Virvou, "A multilayer inference engine for individualized tutoring model: adapting learning material and its granularity," *Neural Computing and Applications*, vol. 35, pp. 61–75, 2023. doi: 10.1007/s00521-021-05740-1.

[12] F. Giannakas, C. Troussas, A. Krouska, C. Sgouropoulou, and I. Voyiatzis, "XGBoost and Deep Neural Network Comparison: The Case of Teams' Performance," in *Intelligent Tutoring Systems (ITS 2021)*, A. I. Cristea and C. Troussas, Eds., Lecture Notes in Computer Science, vol. 12677. Cham: Springer, 2021, pp. 419–426. doi: 10.1007/978-3-030-80421-3_37.

[13] C. Troussas, A. Krouska, and C. Sgouropoulou, "Employing FFNN and Learning Styles to Improve Knowledge Acquisition in Educational Digital Games," in *Novel & Intelligent Digital Systems: Proceedings of the 2nd International Conference (NiDS 2022)*, A. Krouska, C. Troussas, and J. Caro, Eds., Lecture Notes in Networks and Systems, vol. 556. Cham: Springer, 2023, pp. 131–139. doi: 10.1007/978-3-031-17601-2_10.

[14] F. Giannakas, C. Troussas, I. Voyiatzis, and C. Sgouropoulou, "A deep learning classification framework for early prediction of team-based academic performance," *Applied Soft Computing*, vol. 106, p. 107355, 2021. doi: 10.1016/j.asoc.2021.107355.

[15] Nielsen, M., "Neural Networks and Deep Learning", Determination Press, 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com