



ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

BACHELOR THESIS

“WodWide. A social media platform for Crossfit athletes ”

The students

Giapantzis Grigoris
Student ID: it174970

Tolios Dimitris
Student ID: it174904

Supervisor
Diamantaras Konstantinos

26 June 2023

Title: WodWide. A social media platform for Crossfit athletes

Thesis code: 21391

Student's Full Name: Giapantzis Grigoris

Student's Full Name: Tolios Dimitris

Supervisor's Full Name: Diamantaras Kwnstantinos

Date of Thesis Initiation: 10/22/2021

Date of Thesis Completion: 5/26/2023

I certify that I am the author of this thesis and that any assistance I received in preparing it is fully acknowledged and referenced in the thesis. I have also documented any sources from which I have used data, ideas, images, and text, whether they are quoted or paraphrased. Furthermore, I certify that this thesis was prepared by me personally, specifically as a bachelor's thesis, at the Department of Computer Engineering and Electronic Systems of the International University of Greece.

This thesis is the intellectual property of the student Konstantinos Gratsos who completed it. In the context of open access policy, the author/creator grants the International University of Greece a license to reproduce, lend, present to the public, and digitally distribute the thesis internationally, in electronic format and in any medium, for educational and research purposes, without charge. Open access to the full text of the thesis does not in any way imply the transfer of intellectual property rights of the author/creator, nor does it allow reproduction, republication, copying, sale, commercial use, distribution, publication, downloading, uploading, translation, modification in any way, partially or summarily, of the thesis, without the explicit prior written consent of the author/creator.

The approval of the thesis by the Department of Computer Engineering and Electronic Systems of the International University of Greece does not necessarily imply acceptance of the author's views on behalf of the Department.

Abstract

This thesis presents the development of a groundbreaking web application designed specifically for CrossFit athletes, offering a comprehensive suite of features aimed at enhancing their training experience. The app allows users to seamlessly discover and join local gyms, engage in meaningful conversations with fellow gym members, create customized workouts, and leverage artificial intelligence (AI) for accurate repetition verification. To the best of our knowledge, no other existing application combines these diverse functionalities into a single platform tailored for the CrossFit community.

The development process involved extensive research, prototyping, and iterative testing to ensure an intuitive and user-friendly interface. The gym discovery feature integrates location-based services, enabling athletes to easily locate and explore nearby CrossFit facilities, facilitating access to a supportive fitness community. The social interaction component enables athletes to connect, communicate, and share their experiences with like-minded individuals, fostering a sense of camaraderie and motivation. Additionally, the workout creation feature empowers athletes to design personalized training regimens tailored to their specific goals and preferences, providing them with a more personalized and effective training experience.

A key innovation of this web app lies in the incorporation of AI technology for repetition verification. By utilizing computer vision algorithms, the app can accurately analyze and validate exercise repetitions performed by athletes, providing real-time feedback on form and ensuring optimal technique. This unique capability eliminates the need for manual counting or subjective judgment, enhancing both the accuracy and objectivity of repetition tracking, and ultimately promoting safer and more efficient training practices. The experience gained from developing this web app has been highly rewarding. Throughout the process, we honed our technical skills in web development, AI integration, and user experience design. We also gained a deeper understanding of the unique requirements and challenges faced by CrossFit athletes, and the importance of creating a supportive and engaging online environment for their training journeys.

In conclusion, this thesis presents a pioneering web app for CrossFit athletes, revolutionizing the way they discover gyms, interact with fellow athletes, create workouts, and verify repetitions. The combined functionalities offered by the app provide a holistic and empowering training experience, while the incorporation of AI technology enhances accuracy and safety. This project contributes to the advancement of technology in the fitness domain, and we anticipate that it will greatly benefit the CrossFit community and inspire future innovations in the field.

Περίληψη

Αυτό το άρθρο παρουσιάζει μια πρωτοποριακή διαδικτυακή εφαρμογή σχεδιασμένη αποκλειστικά για τους αθλητές CrossFit, με στόχο να ανατρέψει την εμπειρία προπόνησής τους. Το CrossFit είναι ένα εντατικό πρόγραμμα φυσικής άσκησης που συνδυάζει στοιχεία από την αρσεναληπτική άσκηση, την καρδιοαναπνευστική άσκηση και τις λειτουργικές κινήσεις, με σκοπό την προώθηση της συνολικής φυσικής κατάστασης. Η διαδικτυακή εφαρμογή λειτουργεί ως ένα ολοκληρωμένο πλαίσιο που ενσωματώνει την ανακάλυψη γυμναστηρίων, την κοινωνική αλληλεπίδραση, τη δημιουργία προπονήσεων και τον έλεγχο επαναλήψεων με τη χρήση της τεχνητής νοημοσύνης, όλα προσαρμοσμένα αποκλειστικά για τους αθλητές CrossFit.

Η δυνατότητα ανακάλυψης γυμναστηρίων της εφαρμογής χρησιμοποιεί υπηρεσίες βασισμένες στην τοποθεσία, επιτρέποντας στους χρήστες να εντοπίζουν και να εξερευνούν εύκολα γυμναστήρια CrossFit στην περιοχή τους. Αυτή η δυνατότητα απλοποιεί όχι μόνο τη διαδικασία εύρεσης ενός κατάλληλου γυμναστηρίου, αλλά και διευκολύνει την ανάπτυξη μιας υποστηρικτικής κοινότητας φυσικής κατάστασης. Συνδέοντας τους αθλητές με άτομα που έχουν τα ίδια ενδιαφέροντα, η εφαρμογή προάγει την αίσθηση της συντροφικότητας και της κίνητρο, ενισχύοντας τη συνολική εμπειρία προπόνησης των χρηστών.

Επιπλέον, η διαδικτυακή εφαρμογή επιτρέπει στους χρήστες να δημιουργούν προσαρμοσμένες προπονήσεις βασισμένες στους συγκεκριμένους στόχους και τις προτιμήσεις τους. Οι αθλητές μπορούν να σχεδιάζουν προγράμματα εκγύμνασης που προσαρμόζονται στις ατομικές τους ανάγκες, είτε επιδιώκουν βελτίωση της δύναμης, αντοχής ή γενικής απόδοσης. Αυτή η δυνατότητα παρέχει μια πιο ατομική και αποτελεσματική προσέγγιση στην προπόνηση, επιτρέποντας στους αθλητές να βελτιστοποιήσουν τις προπονήσεις τους και να επιτύχουν καλύτερα αποτελέσματα.

Μία από τις βασικές καινοτομίες αυτής της διαδικτυακής εφαρμογής είναι ο έλεγχος επαναλήψεων με τη χρήση της τεχνητής νοημοσύνης. Χρησιμοποιώντας αλγόριθμους υπολογιστικής όρασης, η εφαρμογή μπορεί να αναλύει και να επιβεβαιώνει με ακρίβεια τις επαναλήψεις ασκήσεων που πραγματοποιούν οι αθλητές. Η πραγματικού χρόνου ανατροφοδότηση για τη φόρμα και την τεχνική των ασκήσεων εξασφαλίζει τη βέλτιστη απόδοση και μειώνει τον κίνδυνο τραυματισμού. Αποκλείοντας την ανάγκη για χειροκίνητη καταμέτρηση ή υποκειμενική κρίση, η εφαρμογή προάγει την ακρίβεια και την αντικειμενικότητα στην καταγραφή των επαναλήψεων, ενισχύοντας την ασφάλεια και την αποδοτικότητα της προπόνησης.

Συνοψίζοντας, το άρθρο αυτό παρουσιάζει μια πρωτοποριακή διαδικτυακή εφαρμογή που συνδυάζει διάφορα χαρακτηριστικά για να ανταποκριθεί στις μοναδικές ανάγκες των αθλητών CrossFit. Οι ενσωματωμένες λειτουργίες ανακάλυψης γυμναστηρίων, κοινωνικής αλληλεπίδρασης, δημιουργίας προπονήσεων και έλεγχου επαναλήψεων με τη χρήση της τεχνητής νοημοσύνης δημιουργούν μια ολιστική και ενδυναμωτική εμπειρία προπόνησης. Η ανάπτυξη αυτής της εφαρμογής συνεισφέρει όχι μόνο στην πρόοδο της τεχνολογίας στον τομέα της φυσικής κατάστασης, αλλά και ενισχύει το ταξίδι προπόνησης των αθλητών CrossFit, παρέχοντας μια ολοκληρωμένη και φιλική πλατφόρμα. Με καμία άλλη υπάρχουσα εφαρμογή που να προσφέρει μια τέτοια συνδυασμένη παροχή λειτουργιών, αυτή η διαδικτυακή εφαρμογή ξεχωρίζει ως ένα επαναστατικό εργαλείο για την κοινότητα CrossFit, εμπνέοντας μελλοντικές καινοτομίες στον τομέα.

«WodWide. A social media platform for Crossfit athletes»

«Giapantzis Grigoris, Tolios Dimitris»

Acknowledgments

Firstly, I would like to express my gratitude to Mr. Diamantaras and Mr. Ougiaroglou for their trust in us to undertake and complete this thesis. I appreciate the time they dedicated and their guidance, willingness, and support throughout the writing process, as they were instrumental to its completion. Additionally, I would like to thank my family for their unwavering support throughout my studies.

Contents

Abstract	ii
Περίληψη	iii
Acknowledgments	iv
Contents	v
1 Introduction	1
1.1 Crossfit	1
1.2 Motivation	2
1.3 Contribution	3
1.4 Thesis organization	4
2 Machine Learning	5
2.1 Neural Networks	5
2.2 Deep learning	6
2.3 CNN	6
2.4 MediaPipe Pose: A Versatile Framework for Human Pose Estimation	7
3 Software architecture	9
3.1 Front-end Development and Coding Standards	9
3.1.1 Vue/Nuxt best practises	9
3.1.2 Server Side Rendered Applications	9
3.1.3 Reusable Code and Components	10
3.1.4 Styling Guides	12
3.1.5 Component Testing and End-to-End (E2E) Testing	13
3.1.6 Agile Tools and Flows	14
4 Technologies and Tools	16
4.1 Design tools	16
4.1.1 Figma	17
4.2 Github tools	18
4.2.1 Github	19
4.2.2 Github Actions	19
4.2.3 Github Boards	20
4.2.4 Github Issues	21
4.3 Programming languages	21
4.3.1 Javascript	21
4.3.2 Typescript	22
4.3.3 Python	23
4.4 Nuxt	25
4.5 Styling	26
4.5.1 Tailwing	26
4.5.2 SCSS	28
4.6 Firebase	30
4.7 FastAPI	30
4.8 Testing tools	32
4.8.1 Nuxt-testing-utils	32
4.8.2 Cypress	33
4.9 Netlify	34
5 WodWide Design	36
5.1 WodWide architcture - UML	36
5.2 Backend	39
5.3 Frontend	42
6 WodWide presentation	42
6.1 As a trainer	52
6.2 As an athlete	56
6.3 Video Evaluation	59
7 Conclusion and future work	60

Chapter 1: Introduction

1.1 Crossfit

CrossFit is a high-intensity fitness program that incorporates various elements from different disciplines, such as weightlifting, cardiovascular exercise, gymnastics, and bodyweight movements. It was developed by Greg Glassman in the early 2000s and has gained significant popularity since then.

The primary goal of CrossFit is to improve overall physical fitness and performance through constantly varied workouts. It aims to enhance ten recognized fitness domains: cardiovascular and respiratory endurance, stamina, strength, flexibility, power, speed, coordination, agility, balance, and accuracy. The philosophy behind CrossFit is to create a well-rounded athlete who excels in all these areas, rather than specializing in one particular aspect of fitness.

The training methodology of CrossFit revolves around the concept of functional movements performed at high intensity. Functional movements are those that mimic natural, real-life movements, such as lifting, pushing, pulling, squatting, and running. CrossFit workouts typically combine these functional movements in various combinations and repetitions to create a broad and inclusive fitness program.

CrossFit workouts, also known as "WODs" (Workout of the Day), are constantly varied, meaning that they change regularly and incorporate a wide range of exercises and equipment. This approach helps prevent adaptation and promotes overall fitness development. WODs often include a mix of weightlifting exercises (such as squats, deadlifts, and cleans), cardiovascular activities (like running, rowing, or biking), and gymnastics movements (such as pull-ups, handstands, and rope climbs).

One of the key principles of CrossFit is intensity. The workouts are designed to be challenging and demanding, pushing participants to their limits. The intensity level is scaled according to individual fitness levels and can be adjusted for beginners, intermediate, and advanced athletes. By pushing the body to work at high intensity, CrossFit aims to achieve greater improvements in strength, endurance, and overall fitness in a shorter period.

CrossFit also emphasizes the importance of community and competition. Many CrossFit gyms, known as "boxes," foster a supportive and inclusive environment where participants encourage and motivate each other. CrossFit has also gained popularity through competitive events such as the CrossFit Games, where elite athletes showcase their skills and fitness levels.

While CrossFit has been praised for its ability to improve overall fitness and provide a sense of community, it has also faced some criticism. The high-intensity nature of the workouts can increase the risk of injury if proper form and technique are not maintained. Additionally, the emphasis on intensity may lead to overtraining if not managed effectively.

In conclusion, CrossFit is a fitness program that combines various exercises and training modalities to improve overall physical fitness and performance. It focuses on functional movements, constantly varied workouts, and high intensity. By incorporating elements from weightlifting, cardiovascular exercise, gymnastics, and bodyweight movements, CrossFit aims to create well-rounded athletes who excel in multiple fitness domains.

1.2 Motivation

The motivation behind the development of our all-in-one web app for CrossFit athletes stemmed from a noticeable gap in the market. We recognized the absence of a comprehensive platform that seamlessly integrated all the essential features required by CrossFit enthusiasts. Existing solutions often offered fragmented functionalities, requiring athletes to navigate between multiple applications or resources to access the tools they needed. This lack of a centralized platform inspired us to create a truly all-in-one solution that would streamline the CrossFit experience.

One of our key motivations was to provide CrossFit athletes with a unified platform that consolidated all the necessary features within a single application. We identified the need for a chat feature that would enable athletes to connect and communicate with fellow gym members. Existing options often involved switching to external messaging apps or relying on separate forums, resulting in a disjointed experience. By integrating a chat feature directly into our app, we aimed to foster a sense of community and create a space where athletes could engage in real-time conversations, seek guidance, share experiences, and support one another—all within a single, easily accessible platform.

Another motivation was to simplify the workout planning process for CrossFit athletes. We recognized the time-consuming nature of researching and designing constantly varied workouts tailored to individual preferences and fitness levels. Athletes often had to rely on external resources or create their own routines, which could be overwhelming and hinder their progress. To address this challenge, we developed a WOD generator application within our app. By leveraging intelligent algorithms, this feature generates personalized workout routines, taking into account an athlete's preferences, strengths, and weaknesses. With the integration of the WOD generator, athletes no longer need to spend valuable time planning their workouts, allowing them to focus on training and achieving their fitness goals.

Competitiveness and motivation are central to the CrossFit ethos. Athletes seek to push their limits, track their progress, and compare their performances with fellow members. However, the lack of a centralized leaderboard system often required athletes to rely on external resources or rely solely on gym-specific mechanisms. To address this, our app incorporates a leaderboard feature, enabling athletes to monitor their rankings and engage in healthy competition within the community. By integrating the leaderboard directly into the app, we provide a seamless and convenient way for athletes to track their progress, set goals, and challenge themselves alongside their peers.

Furthermore, our motivation extended to supporting gym owners within the CrossFit community. Gym owners play a vital role in creating a thriving and engaging fitness environment, yet many faced challenges in disseminating workout information and maintaining consistent communication with their members. To address this, we developed a dedicated workout upload platform for gym owners within our app. This feature empowers gym owners to easily share workout descriptions, scaling options, and video demonstrations, ensuring clear and consistent communication with their members. By providing this platform, we aimed to streamline gym operations, foster effective communication, and contribute to the overall success of CrossFit gyms.

In summary, the motivation behind our all-in-one web app for CrossFit athletes stemmed from the lack of an existing platform that combined all the essential features into a unified solution. By addressing this gap, we aimed to simplify the CrossFit experience, streamline communication, enhance workout planning, foster competitiveness, and support gym owners. Our motivation lay in providing athletes and

gym owners with a comprehensive, user-friendly, and centralized platform that catered to their specific needs, ultimately contributing to a more engaging and successful CrossFit community.

1.3 Contribution

As the popularity of CrossFit continues to grow, the need for a comprehensive and user-friendly digital platform to support CrossFit athletes and gym owners has become increasingly evident. This section explores the motivation behind the development of an all-in-one web app specifically tailored to meet the unique needs of CrossFit enthusiasts. The proposed app aims to provide a seamless and efficient experience by integrating communication features, a WOD generator application, a leaderboard for maintaining competitiveness, and a platform for gym owners to upload workouts. This section elucidates the motivations behind each of these key features and emphasizes their value in enhancing the CrossFit experience.

Enhanced Communication and Community Engagement: One of the primary motivations behind the development of the all-in-one web app for CrossFit athletes is to facilitate effective communication and foster a strong sense of community among gym members. The incorporation of a chat feature within the app allows athletes to engage in real-time conversations, share training tips, motivate each other, and build connections beyond the physical gym space. By promoting interaction and support, the app cultivates a supportive environment that strengthens the bond between CrossFit enthusiasts.

WOD Generator Application: The WOD generator application serves as a vital component of the all-in-one web app, providing athletes with dynamic and varied workout routines. CrossFit training emphasizes the importance of constantly varied workouts, and the WOD generator fulfills this need by generating diverse exercise combinations tailored to individual fitness levels. The app's intelligent algorithm takes into account an athlete's preferences, strengths, and weaknesses to provide personalized and challenging workouts. By eliminating the need for athletes to spend time planning workouts, the app streamlines the training process, allowing individuals to focus on maximizing their performance and achieving their fitness goals.

Leaderboard for Competitive Motivation: Competition is an integral part of CrossFit, and the inclusion of a leaderboard within the app serves to maintain the competitive spirit among athletes. The leaderboard feature enables athletes to compare their performances with other members, track their progress, and strive for continual improvement. By showcasing rankings and achievements, the app incentivizes athletes to push their boundaries, set new goals, and engage in healthy competition within the community. The leaderboard fosters a sense of accountability and motivation, driving individuals to elevate their fitness levels.

Workout Upload Platform for Gym Owners: Recognizing the significance of gym owners in the CrossFit ecosystem, the all-in-one web app provides a dedicated platform for them to upload workouts and engage with their members. This feature empowers gym owners to easily disseminate workout information, including descriptions, scaling options, and video demonstrations, ensuring consistency and quality in training across the gym. Furthermore, gym owners can use the app to communicate important announcements, updates, and event information, thereby strengthening the connection between the gym and its members.

The development of an all-in-one web app for CrossFit athletes arises from the recognition of the unique

requirements of this fitness community. By integrating communication features, a WOD generator application, a leaderboard, and a workout upload platform for gym owners, the app aims to optimize the CrossFit experience, fostering engagement, competition, and personal growth. This comprehensive digital platform empowers athletes to communicate, train effectively, monitor progress, and connect with their gym community, ultimately enhancing their overall CrossFit journey.

1.4 Thesis organization

The thesis consists of 7 chapters. So far, there has been a brief mention of what will follow, as well as an introduction of Crossfit.

Chapter 2 will provide a detailed description of something about machine learning or AI. Who knows

Chapter 3 focuses on the software architecture employed in this study. It presents a comprehensive examination of the underlying principles and methodologies guiding the development process. The chapter delves into the establishment of coding standards, the systematic organization of code components, and the effective utilization of development tools.

Chapter 4 presents a detailed analysis of the technologies employed in the development of the application. It provides an in-depth examination of the tools, software development kits (SDKs), libraries, and frameworks utilized throughout the project.

Chapter 5 delves into the WodWide design. This chapter encompasses an in-depth analysis of the web applications' architecture using Unified Modeling Language (UML), providing a visual representation of the various components and their interactions. Furthermore, it elucidates the design considerations and decisions made in developing the backend and frontend components of the application.

Chapter 6 offers a comprehensive showcase of the application's website, accompanied by insightful case studies. This chapter presents a detailed examination of the website's user interface, navigation, and overall user experience. Moreover, this chapter presents case studies that illustrate real-world scenarios and highlight the practical application and effectiveness of the website in addressing specific user needs and achieving desired outcomes.

Chapter 7, the final chapter, presents concluding remarks, research insights, and future extensions of the web app. It summarizes key findings, implications, and user satisfaction. It discusses potential enhancements, innovative features, and emerging trends. This chapter provides a foundation for future research and development, encouraging continuous innovation and technological competitiveness.

Chapter 2: Machine Learning

2.1 Neural Networks

In recent years, neural networks have experienced a renaissance, fueled by advancements in computational power, the availability of vast datasets, and breakthroughs in optimization algorithms. These factors have enabled the training of deeper and more complex neural networks, unlocking their potential to learn from massive amounts of data and solve highly intricate tasks. As a result, neural networks have achieved state-of-the-art performance in various domains, ranging from image recognition and natural language processing to robotics and autonomous systems.

At its core, a neural network is composed of interconnected artificial neurons, known as nodes or units, organized into layers. The input layer receives raw data, which is then processed through one or more hidden layers, and ultimately produces an output layer that represents the network's prediction or classification. The connections between neurons are governed by learnable weights, which allow the network to adapt and optimize its performance during training. [1]

Neural networks operate by propagating data forward through the layers, a process known as feedforward computation. Each neuron receives input signals, applies an activation function to produce an output, and passes it to the subsequent layer. Through this interconnected structure, neural networks are capable of learning hierarchical representations of the input data, capturing intricate features and relationships.

The success of neural networks heavily relies on their ability to learn from data. During the training phase, the network is presented with a labeled dataset, and its parameters, including the weights, are adjusted to minimize a predefined loss function. This adjustment is performed through an optimization algorithm, often based on gradient descent, which computes the gradients of the loss with respect to the network's parameters. [2]

The backpropagation algorithm is a key technique for efficiently calculating these gradients. It propagates the errors backwards from the output layer to the hidden layers, allowing for the adjustment of weights that contribute most significantly to the overall prediction error. This iterative process of updating weights based on gradients gradually improves the network's performance, leading to more accurate predictions and classifications. [3]

Neural networks have found extensive applications in academic research across numerous disciplines. In computer vision, convolutional neural networks (CNNs) have revolutionized image recognition, object detection, and image generation tasks. Recurrent neural networks (RNNs) and their variants have demonstrated remarkable success in natural language processing tasks, including language translation, sentiment analysis, and text generation. Furthermore, neural networks have been employed in fields such as bioinformatics, finance, social sciences, and physics, enabling breakthroughs in data analysis, pattern recognition, and predictive modeling. [4]

The versatility and flexibility of neural networks make them a compelling choice for academic researchers tackling complex problems and analyzing large datasets. Their ability to capture intricate patterns, learn from raw data, and generalize to unseen examples has significantly advanced the boundaries of knowledge in various domains.

2.2 Deep learning

Deep learning, a subfield of machine learning, has emerged as a powerful approach for modeling and understanding complex patterns in data. With its ability to automatically learn hierarchical representations from raw input, deep learning has revolutionized various domains, including computer vision, natural language processing, and speech recognition. The key characteristic of deep learning lies in the utilization of deep neural networks, typically composed of multiple layers, to extract intricate features and capture nonlinear relationships within the data. These networks have demonstrated exceptional performance in tasks such as image classification, object detection, language translation, and speech synthesis. However, the success of deep learning also brings challenges, such as the need for large labeled datasets, computational resources, and interpretability of complex models. This research paper aims to explore the advancements, applications, and challenges in deep learning, providing insights into its current state and future directions for further research and development.

2.3 CNN

Convolutional Neural Networks (CNNs) are a specialized type of neural network architecture designed for processing and analyzing structured grid-like data, particularly images or other grid-like data such as audio spectrograms. CNNs have achieved remarkable success in various computer vision tasks, including image classification, object detection, and image segmentation. [5]

CNNs are specifically designed to exploit the spatial relationships present in image data. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers.

The key component of a CNN is the convolutional layer, which applies a set of learnable filters (also known as kernels) to the input image. Each filter slides over the input image, computing a dot product between its weights and the corresponding local receptive field. This process results in a feature map that captures important visual patterns or features at different spatial locations. The convolutional layer typically has multiple filters, allowing it to extract multiple feature maps simultaneously.

Pooling layers are often inserted after convolutional layers to reduce the spatial dimensions of the feature maps while retaining their essential information. Max pooling is a commonly used pooling technique, where the feature map is divided into non-overlapping regions, and the maximum value within each region is retained. This downsampling operation helps to make the network more robust to variations in the input data and reduces the number of parameters in the subsequent layers.

After several convolutional and pooling layers, the output is flattened and passed through one or more fully connected layers. These layers are similar to those found in traditional neural networks and are responsible for learning higher-level representations and making final predictions.

Training a CNN involves initializing the network with random weights and optimizing them through a process called backpropagation. Backpropagation calculates the gradients of the network's parameters with respect to a defined loss function, allowing the weights to be adjusted in the direction that minimizes the loss. Common optimization algorithms used for training CNNs include stochastic gradient descent (SGD) and its variants, such as Adam and RMSprop.

CNNs have shown great success in a wide range of computer vision tasks due to their ability to automati-

cally learn hierarchical representations from raw image data. They can capture local patterns in the early layers and gradually learn more complex and abstract features in deeper layers. Additionally, CNNs can be pretrained on large-scale datasets, such as ImageNet, and fine-tuned on specific tasks with smaller datasets, enabling transfer learning and improving performance on limited data.

In recent years, CNNs have also been extended to handle other types of structured data, such as 3D data in medical imaging or point cloud data in 3D object recognition, through architectures like 3D CNNs and PointNet. These advancements have further expanded the applications of CNNs beyond traditional image processing tasks.

2.4 MediaPipe Pose: A Versatile Framework for Human Pose Estimation

In recent years, human pose estimation has emerged as a fundamental task in computer vision and has found numerous applications in areas such as activity recognition, human-computer interaction, and augmented reality. With the increasing availability of powerful deep learning techniques, various pose estimation frameworks have been developed to accurately and efficiently infer the human body's skeletal structure from images or videos. One such prominent framework is MediaPipe Pose, developed by Google.

MediaPipe Pose is an open-source cross-platform framework that offers real-time, high-fidelity 2D and 3D pose estimation capabilities. Leveraging deep learning models and efficient inference algorithms, MediaPipe Pose enables robust pose estimation on a wide range of devices, including smartphones, laptops, and embedded systems. This framework serves as a valuable tool for researchers and developers seeking to explore and integrate human pose estimation in their applications. [6]

At the core of MediaPipe Pose lies a deep learning-based pose estimation model trained on extensive annotated datasets. This model utilizes convolutional neural networks (CNNs) to extract meaningful spatial features from input images or frames. The architecture of the model comprises several stacked convolutional layers followed by intermediate connection modules to capture multi-scale information. The final output of the model is a set of keypoint heatmaps, where each heatmap represents the likelihood of the presence of a specific body joint.

To achieve real-time performance, MediaPipe Pose leverages efficient inference techniques such as single-shot detection and keypoint association. These techniques allow the model to process input data in a streaming manner, reducing computational overhead and ensuring low-latency pose estimation. Additionally, MediaPipe Pose incorporates temporal smoothing and tracking algorithms to enhance the robustness and temporal coherence of the pose estimation output, particularly in challenging scenarios involving occlusions or rapid motion.

MediaPipe Pose offers a wide range of features and capabilities that make it a versatile framework for human pose estimation. Firstly, the framework supports both 2D and 3D pose estimation, allowing researchers to choose the appropriate representation based on their specific application requirements. This flexibility enables the estimation of joint positions in image coordinates or their corresponding 3D coordinates in real-world space.

Furthermore, MediaPipe Pose provides a rich set of pre-trained models for different use cases. These models are trained on diverse datasets, including various activities and body types, ensuring generaliza-

tion across different scenarios. Moreover, the framework supports model customization and fine-tuning, allowing researchers to adapt the models to specific domains or datasets.

Additionally, MediaPipe Pose integrates seamlessly with the wider MediaPipe ecosystem, which offers a range of additional modules for audio, video, and gesture recognition. This integration enables researchers to build complex multimodal applications by combining pose estimation with other MediaPipe functionalities.

MediaPipe Pose has been widely adopted in both academia and industry, fueling advancements in diverse domains. Researchers have utilized this framework for applications such as human action recognition, pose-based human-computer interaction, sports analytics, and virtual reality. Its real-time performance, accuracy, and versatility have made it a preferred choice for many projects seeking to leverage human pose estimation.

Moreover, the open-source nature of MediaPipe Pose has fostered a collaborative community of researchers and developers, facilitating knowledge exchange and the development of new techniques and applications. The availability of pre-trained models, extensive documentation, and code samples has accelerated the adoption of MediaPipe Pose in both research and industry settings.

In conclusion, Google's MediaPipe Pose provides a powerful and versatile framework for human pose estimation, offering real-time, high-fidelity 2D and 3D pose estimation capabilities. Its architecture, efficient inference techniques, and integration with the broader MediaPipe ecosystem make it an ideal tool for researchers and developers working on diverse applications that require robust and accurate human pose estimation. The framework's impact on the research community and its adoption in real-world applications highlight its effectiveness and value in advancing the field of human pose estimation.



Figure 2.1: Media Pipe

Chapter 3: Software architecture

This chapter aims to provide an overview of our organization's coding standards, best practices, and guidelines related to front-end development, server-side rendered applications, reusable code and components, styling guides, testing and agile tools and flows.

3.1 Front-end Development and Coding Standards

Front-end development plays a pivotal role in crafting user interfaces that are intuitive, visually appealing, and responsive. To ensure consistency and maintainability across projects, our organization adheres to a set of coding standards for front-end development. These standards encompass naming conventions, code organization, file structure, and documentation practices. By following these standards, developers can seamlessly collaborate on front-end codebases, enhance code readability, and facilitate future maintenance and updates.

3.1.1 Vue/Nuxt best practises

- **Vue.js Style Guide:** The official Vue.js style guide, maintained by the core team, provides a comprehensive set of best practices for writing Vue code. It covers everything from component naming conventions to indentation, state management, and more. [7]
- **Vue.js Component Style Guide:** This guide focuses specifically on writing reusable and maintainable Vue components. It covers component structure, props, events, slots, and other best practices. [8]
- **ESLint Vue Plugin:** ESLint is a popular JavaScript linter that helps enforce coding standards. The ESLint Vue Plugin provides additional rules specifically for Vue.js projects, ensuring consistent code across your Vue components. [9]
- **Vue Patterns:** This website collects various patterns and best practices for Vue.js development. It covers topics such as component communication, state management, routing, and more. [10]

3.1.2 Server Side Rendered Applications

Server-side rendering (SSR) refers to the process of generating the final HTML content of a web page on the server and sending it to the client's browser. This approach is chosen for applications that prioritize search engine optimization (SEO) because it offers several benefits in terms of visibility and discoverability by search engines. Here's why we might have chosen SSR for our application and how it helps with SEO:

- **Search engine crawler compatibility:** Many search engine crawlers are proficient at parsing and understanding HTML content. By serving fully rendered HTML pages, SSR ensures that search engine bots can easily crawl and index the content. This is in contrast to purely client-side rendered applications, where search engines may have difficulties extracting meaningful information from JavaScript-heavy code.

- **Improved initial load time:** When a user visits a website, the initial load time is crucial for their experience and can impact SEO rankings. SSR allows us to send pre-rendered HTML content to the user's browser, reducing the time needed to display the page. This approach provides faster perceived performance compared to client-side rendering, where the browser must wait for JavaScript to be downloaded, executed, and dynamically render the content.
- **Enhanced metadata and structured data:** SEO relies on various metadata elements and structured data markup. With SSR, we can ensure that the initial HTML response includes all the necessary metadata, such as title tags, meta descriptions, and Open Graph tags. Additionally, we can include structured data markup, such as Schema.org annotations, directly in the HTML, making it easier for search engines to understand and index the content accurately.
- **Social media sharing:** When users share links to our application on social media platforms, SSR plays a crucial role in optimizing the shared content. By providing fully rendered HTML pages, we can control the information displayed in link previews, such as titles, descriptions, and images. This control over shared content can lead to better engagement, increased click-through rates, and improved SEO indirectly.
- **Dynamic content indexing:** While SSR primarily generates static HTML pages, it can also incorporate dynamic content during the rendering process. By fetching data from APIs or databases on the server before rendering, SSR enables search engines to index dynamic content effectively. This ensures that search engine results reflect the most up-to-date information from our application.
- **Compatibility with older browsers and systems:** Some users may have older browsers or systems that lack robust JavaScript support. SSR ensures that these users can still access and navigate our application effectively, improving accessibility and potentially attracting a broader audience.

It's important to note that while SSR offers SEO benefits, it also introduces additional complexity in terms of server-side rendering configuration, caching strategies, and potential performance trade-offs. Careful consideration should be given to factors like application complexity, development resources, and the specific SEO requirements before deciding to adopt SSR for an application.

3.1.3 Reusable Code and Components

In modern software development, the concept of reusable code and components plays a crucial role in improving efficiency, maintainability, and scalability of software systems. Reusability allows developers to leverage existing code and components to create new applications, reducing development time and effort. This chapter explores the benefits of reusable code and components and delves into the logic behind their implementation, highlighting the importance of design principles, modularization, and abstraction.

Benefits of Reusable Code and Components

- **Improved Efficiency.** One of the primary advantages of reusable code and components is improved efficiency. By reusing existing code, developers can avoid reinventing the wheel and leverage solutions that have already been tested and proven. This reduces development time significantly, allowing developers to focus on implementing unique features and functionalities specific to their

application. Moreover, reusable components can be optimized and fine-tuned over time, leading to enhanced performance and reduced resource consumption.

- **Enhanced Maintainability.** Reusable code and components promote maintainability by encapsulating functionality into self-contained units. When changes or bug fixes are required, developers can make modifications in a centralized location, eliminating the need for redundant changes across multiple applications. This centralized approach simplifies maintenance efforts, improves code consistency, and reduces the chances of introducing errors during updates. Additionally, reusable components often undergo rigorous testing and quality assurance processes, contributing to improved overall system stability.
- **Scalability and Flexibility.** Reusable code and components facilitate scalability by enabling developers to easily extend existing functionality or integrate new features. As systems grow and evolve, modular components can be adapted or combined to fulfill changing requirements. This flexibility allows developers to respond efficiently to evolving user needs and business demands. Furthermore, reusable components enable organizations to build a library of standardized modules, fostering consistency and facilitating collaboration among development teams.
- **Cost-Effectiveness.** Reusability can lead to significant cost savings in software development projects. By leveraging existing code and components, organizations can reduce development time and effort, resulting in decreased project costs. Additionally, reusable components can be licensed or shared across projects, eliminating the need for redundant development or costly third-party solutions. These cost savings can be allocated to other areas of the project, such as user experience enhancements or further research and development.

Logic behind Reusable Code and Components

- **Design Principles.** The logic behind reusable code and components is rooted in the application of sound design principles. Modularization and encapsulation are essential principles that contribute to reusability. Breaking down a system into modular components that perform specific tasks or provide distinct functionalities allows for easier comprehension, maintenance, and reusability. By encapsulating related functionality within a component, developers can create self-contained units that are independent of the rest of the system, promoting flexibility and reducing interdependencies.
- **Abstraction.** Abstraction is another key aspect of building reusable code and components. By abstracting the underlying implementation details, developers can expose a simplified interface that hides complexity and allows users to interact with the component at a higher level of abstraction. This separation of concerns between the interface and the implementation promotes code reuse, as components can be used without requiring in-depth knowledge of their internal workings. Abstraction also facilitates the modification or replacement of underlying implementations without affecting the functionality of the component.
- **Documentation and Standardization.** Effective documentation and standardization practices further enhance the logic behind reusable code and components. Well-documented code and component libraries provide clear instructions on how to use and integrate them into different projects. This documentation includes details on the purpose, functionality, input/output parameters, and usage examples, making it easier for developers to understand and utilize the components effectively.

Standardization of coding practices, naming conventions, and component interfaces ensures consistency and seamless integration across different projects.

3.1.4 Styling Guides

BEM guide

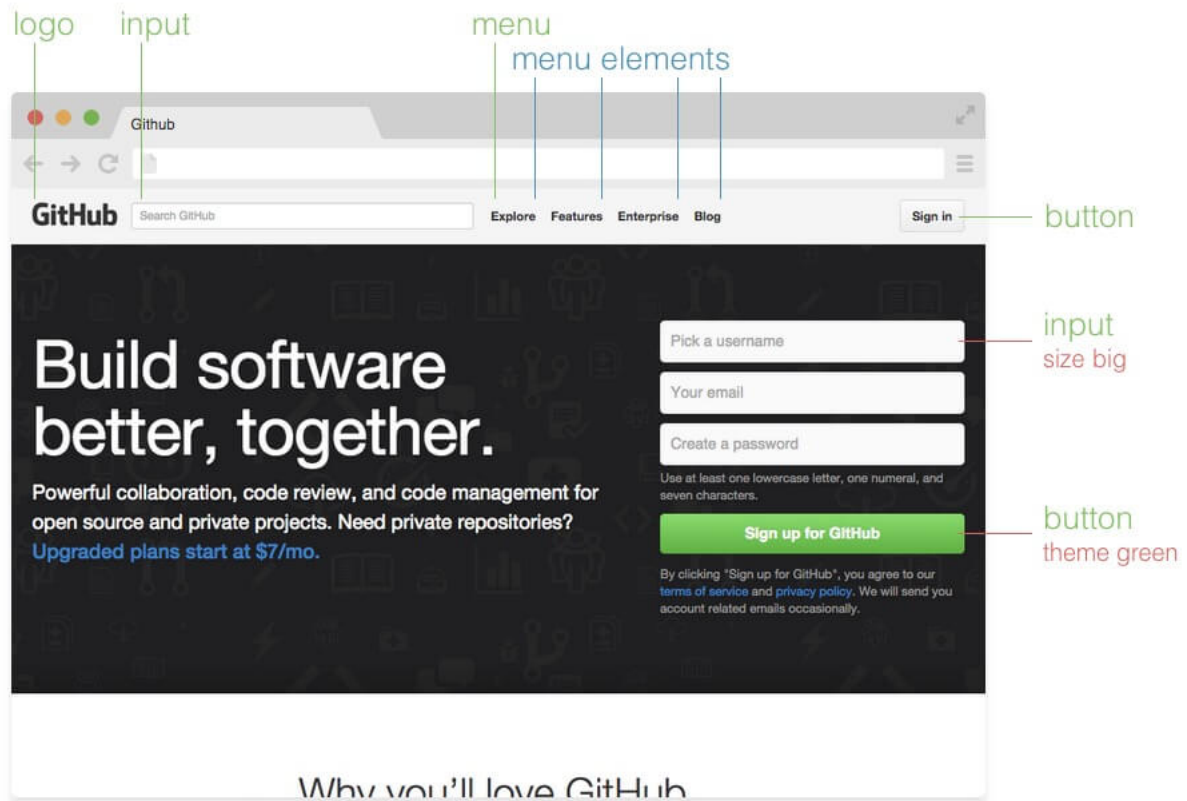


Figure 3.1: BEM guide

BEM-like Naming

- `person {}`
- `person__head {}`
- `person--tall {}`

Elements are delimited with two (2) underscores (`__`), and Modifiers are delimited by two (2) hyphens (`--`). Here we can see that `.person {}` is the Block; it is the sole root of a discrete entity. `.person__head {}` is an Element; it is a smaller part of the `.person {}` Block. Finally, `.person--tall {}` is a Modifier; it is a specific variant of the `.person {}` Block.

More Layers

If we were to add another Element—called, let's say, `.person__eye {}`—to this `.person {}` component, we would not need to step through every layer of the DOM. That is to say, the correct notation

would be `.person__eye {}`, and not `.person__head__eye {}`. Your classes do not reflect the full paper-trail of the DOM.

Modifying Elements

You can have variants of Elements, and these can be denoted in a number of ways depending on how and why they are modified

- `person__eye-blue {}`

Naming Conventions in HTML

Naming conventions aren't necessarily all that useful in your CSS. Where naming conventions' power really lies is in your markup. Take the following, non-naming-conventioned HTML:

```
<div class="box profile pro-user">
  <img class="avatar image">
  <p class="bio">
    ...
  </p>
</div>
```

How are the classes `box` and `profile` related to each other? How are the classes `profile` and `avatar` related to each other? Are they related at all? Should you be using `pro-user` alongside `bio`? Will the classes `image` and `profile` live in the same part of the CSS? Can you use `avatar` anywhere else?

From that markup alone, it is very hard to answer any of those questions. Using a naming convention, however, changes all that:

```
<div class="box profile profile--is-pro-user">
  <img class="avatar profile__image" />
  <p class="profile__bio">
    ...
  </p>
</div>
```

Now we can clearly see which classes are and are not related to each other, and how; we know what classes we can't use outside of the scope of this component; and we know which classes we may be free to reuse elsewhere. [11]

3.1.5 Component Testing and End-to-End (E2E) Testing

Testing is an integral part of the software development lifecycle, ensuring the quality and reliability of software systems. Component testing and end-to-end (E2E) testing are two essential testing methodologies that play a vital role in verifying the functionality and performance of software components and

entire systems. This chapter explores the benefits of component testing and E2E testing and delves into the logic behind their implementation, highlighting the importance of test coverage, test automation, and test data management.

Component Testing Benefits Component testing, also known as unit testing, focuses on verifying the correctness of individual software components in isolation. The benefits of component testing include:

- **Early Bug Detection:** Component testing enables early bug detection by catching defects at the unit level. It helps identify and fix issues before they propagate to other parts of the system, reducing the overall debugging effort and improving software quality.
- **Faster Debugging:** By testing components in isolation, developers can narrow down the source of errors more easily. This targeted approach accelerates the debugging process and allows for quicker resolution of issues.
- **Facilitates Refactoring:** Component testing provides confidence during code refactoring by ensuring that existing functionality remains intact. Tests act as safety nets, allowing developers to make changes without fear of breaking the system.
- **Documentation and Examples:** Well-written component tests serve as documentation and examples of how to use and interact with components. They provide usage scenarios and usage guidelines, making it easier for other developers to understand the intended functionality.

E2E Testing Benefits E2E testing focuses on testing the entire software system or specific user workflows to ensure that all components work together as expected. The benefits of E2E testing include:

- **Integration Verification:** E2E testing validates the integration and interaction of different components, ensuring that they work harmoniously to deliver the desired functionality. It helps identify issues that may arise due to component interactions, such as data inconsistencies or communication failures.
- **User-Centric Validation:** E2E testing simulates real-world user scenarios, validating the system from the user's perspective. It ensures that the software meets user expectations and functions correctly in different usage scenarios, resulting in an improved user experience.
- **Error Detection in Complex Workflows:** E2E testing helps uncover errors that may arise in complex workflows involving multiple components. It verifies the correctness of data flow, state transitions, and business logic, ensuring that the system behaves as intended in end-to-end scenarios.
- **Performance and Scalability Testing:** E2E testing provides an opportunity to evaluate the performance and scalability of the entire system. It helps identify bottlenecks, performance degradation, or resource limitations that may occur under real-world usage conditions.

3.1.6 Agile Tools and Flows

Agile methodologies have revolutionized software development by emphasizing iterative development, adaptability, and collaboration. Agile tools and flows play a crucial role in enabling effective implementation of agile principles and practices. This chapter explores the benefits of utilizing agile tools and flows

in software development projects and delves into the logic behind their implementation, highlighting the importance of transparency, communication, and continuous improvement.

Benefits of Agile Tools and Flows

- **Enhanced Collaboration.** One of the primary advantages of agile tools and flows is the facilitation of enhanced collaboration among team members. Agile methodologies promote cross-functional teams and active stakeholder involvement. Agile tools provide platforms for communication, task tracking, and knowledge sharing, enabling teams to collaborate seamlessly. By promoting transparency and real-time updates, these tools foster effective collaboration, leading to improved teamwork, faster decision-making, and the ability to respond quickly to changing requirements.
- **Improved Visibility and Transparency.** Agile tools and flows offer improved visibility into project progress, enabling teams to track and monitor tasks, user stories, and project milestones. Visual dashboards, such as Kanban boards or burndown charts, provide real-time updates on work in progress, bottlenecks, and overall project status. This transparency allows stakeholders to have a clear understanding of the project's current state, fosters accountability, and helps identify potential issues or risks at an early stage. Improved visibility also facilitates effective communication and collaboration within the team and with external stakeholders.
- **Adaptability and Flexibility.** Agile tools and flows promote adaptability and flexibility, enabling teams to respond to changing requirements and market conditions. Agile methodologies encourage iterative development and frequent feedback loops. Tools such as user story management systems or backlog prioritization tools facilitate the quick reordering of tasks or user stories based on changing priorities. This adaptability allows teams to deliver incremental value and respond to evolving customer needs, ensuring that the final product aligns with the stakeholders' expectations.
- **Continuous Improvement.** Agile tools and flows support the concept of continuous improvement by providing mechanisms for retrospectives, feedback gathering, and process optimization. Retrospective tools enable teams to reflect on past iterations, identify areas of improvement, and implement changes in subsequent iterations. Feedback gathering tools, such as surveys or feedback loops, allow stakeholders to provide input and suggestions, leading to continuous refinement of the product. By leveraging these tools, teams can foster a culture of learning, innovation, and continuous improvement.

Logic behind Agile Tools and Flows

- **Transparency and Communication.** The logic behind agile tools and flows is rooted in the principles of transparency and communication. Agile methodologies emphasize open and transparent communication among team members and stakeholders. Tools such as project management software, collaboration platforms, and video conferencing tools facilitate real-time communication, ensuring that all team members have access to relevant information and can contribute effectively. Transparent tools also promote accountability and trust, as they provide visibility into each team member's contributions and progress.
- **Iterative Development and Feedback Loops.** Agile tools and flows support the iterative development approach, enabling teams to deliver value incrementally and gather feedback early in the

process. Task tracking tools, user story management systems, and feedback gathering mechanisms enable teams to break down work into manageable increments, prioritize tasks, and gather feedback from stakeholders. These tools facilitate the implementation of short feedback loops, allowing teams to incorporate feedback and make adjustments promptly, resulting in a product that better meets customer expectations.

- **Continuous Monitoring and Improvement.** Agile tools and flows enable continuous monitoring of project progress, team performance, and process efficiency. Tools such as burndown charts, velocity trackers, and retrospective facilitation tools help teams evaluate their performance and identify areas for improvement. By collecting and analyzing data on a regular basis, teams can make data-driven decisions, optimize their processes, and continually enhance their performance and productivity.

Chapter 4: Technologies and Tools

4.1 Design tools

The design phase of developing a web application plays a crucial role in creating an intuitive and visually appealing user interface. Design tools empower developers and designers to bring their ideas to life, facilitating the creation of engaging and user-friendly experiences. In this subchapter, we explore the design tools employed in the development process of our all-in-one platform for CrossFit athletes.

Effective design tools offer a range of features and functionalities that streamline the design workflow and enable collaborative work among team members. From wireframing and prototyping to creating high-fidelity designs, these tools assist in visualizing concepts, iterating designs, and ultimately delivering a polished user interface.

In this section, we discuss the design tools utilized throughout the development of our web application. We delve into the selection process, highlighting the key considerations that influenced our choices. Furthermore, we outline the specific features and benefits of each tool, showcasing how they contributed to the overall design process.

By employing the right design tools, we aimed to ensure the creation of a visually captivating and user-centric interface for our all-in-one platform. Through a systematic evaluation of design tools, we were able to leverage their capabilities and optimize the design workflow, leading to an enhanced user experience.

In the subsequent subsections, we provide an overview of the design tools used, detailing their functionalities, advantages, and contributions to the development process. By exploring these design tools, we gain insight into how they shaped our web application's visual aesthetics and usability.

Note: Make sure to adapt and expand this introduction according to the specific design tools you used and their significance in your project.

4.1.1 Figma

Figma: Streamlining Collaborative Design

One of the key factors that influenced our choice of design tool was the need for a collaborative platform that could streamline the design process and facilitate effective communication among team members. After careful evaluation, we selected Figma as our primary design tool due to its powerful collaborative features and versatility.

- **Cloud-Based Collaboration:** Figma's cloud-based nature allowed our design team to collaborate seamlessly in real-time. Multiple team members could work on the same design file simultaneously, eliminating the need for cumbersome file sharing and version control. This collaborative environment promoted efficient teamwork, enabling designers, developers, and stakeholders to provide feedback and iterate designs swiftly.
- **Cross-Platform Accessibility:** Figma's cross-platform compatibility was a significant advantage for our team. With Figma's web-based application and desktop clients for Windows and macOS, our designers could work on their preferred operating systems without any compatibility issues. This flexibility ensured that the design process was not hindered by platform constraints and facilitated a more inclusive and efficient workflow.
- **Design Components and Libraries:** Figma's robust design component system allowed us to create reusable UI elements, ensuring design consistency across the application. By utilizing components, we could update a single instance, and the changes would be reflected throughout the entire design, reducing redundancy and saving valuable design time. Additionally, Figma's extensive library of pre-built UI components and plugins further accelerated the design process, enabling rapid prototyping and iteration.
- **Prototyping and Interactions:** Figma's prototyping capabilities played a vital role in visualizing and validating the user experience. With its intuitive interface, we could create interactive prototypes, define user flows, and simulate user interactions. This functionality enabled us to conduct usability testing, gather valuable feedback, and iterate on the design before development, resulting in a more refined and user-friendly interface.
- **Developer Handoff:** Figma excelled in its ability to facilitate a smooth handoff from design to development. Through its developer handoff feature, we could generate design specifications, export assets, and provide detailed design annotations directly within the tool. This streamlined communication between designers and developers, ensuring accurate implementation of the design and reducing the likelihood of misunderstandings or inconsistencies during the development phase.

In summary, Figma emerged as the ideal design tool for our web application due to its collaborative features, cross-platform accessibility, design component system, prototyping capabilities, and developer handoff functionality. By utilizing Figma, we were able to foster effective collaboration, iterate designs efficiently, and deliver a visually consistent and user-centric interface for our all-in-one platform for CrossFit athletes.

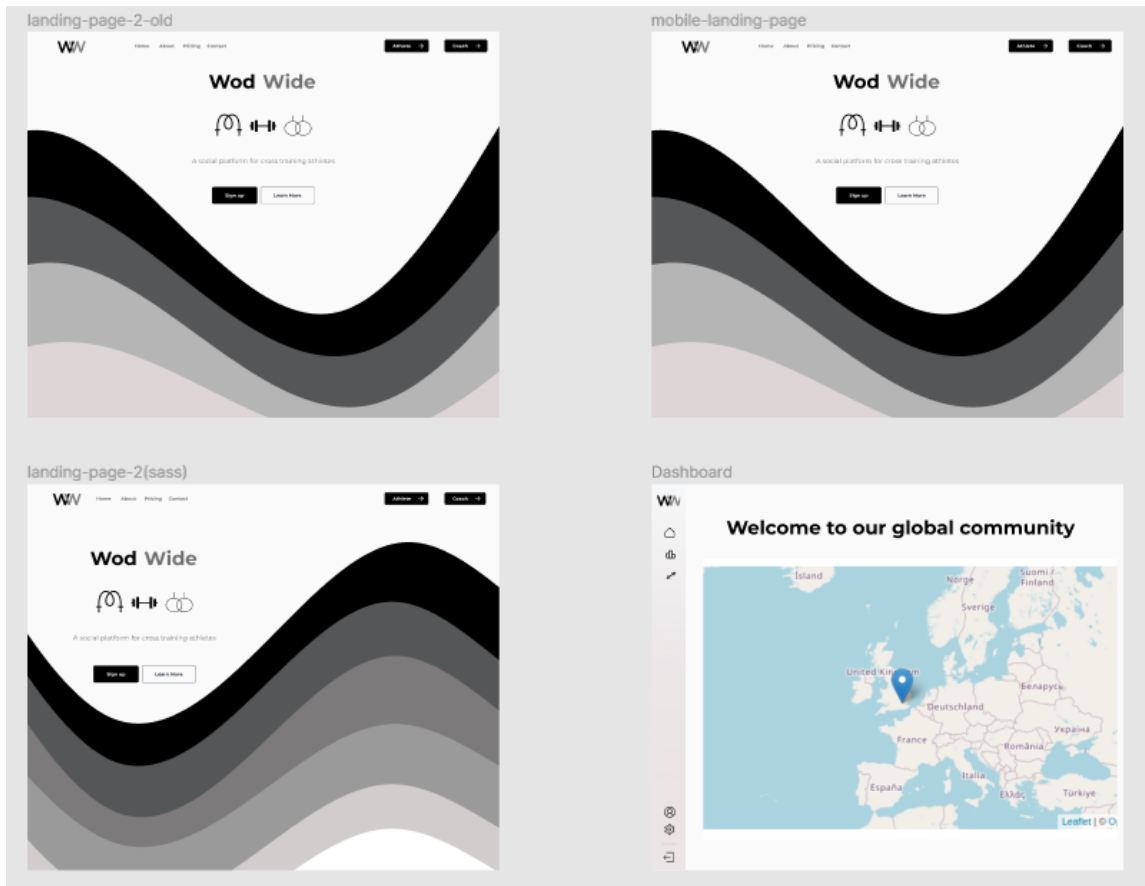


Figure 4.1: Figma desktop mockups



Figure 4.2: Figma mobile mockups

4.2 Github tools

GitHub has revolutionized software development practices by providing a comprehensive platform for version control, collaboration, and project management. As a widely used Git hosting service, GitHub offers an array of tools and features that streamline development workflows, enhance team collaboration, and facilitate efficient project management. In this section, we explore the key aspects of GitHub and its

associated tools, including GitHub Actions, GitHub Boards, and GitHub Issues.

4.2.1 Github

At the core of GitHub's offerings is its role as a Git hosting service. Developers can leverage GitHub to store, manage, and track changes to their codebase using the Git version control system. By utilizing GitHub's distributed architecture, developers can collaborate seamlessly, ensuring that everyone on the team has access to the latest codebase and its history. Additionally, GitHub provides a user-friendly web interface that simplifies repository management and supports seamless integration with various development tools.

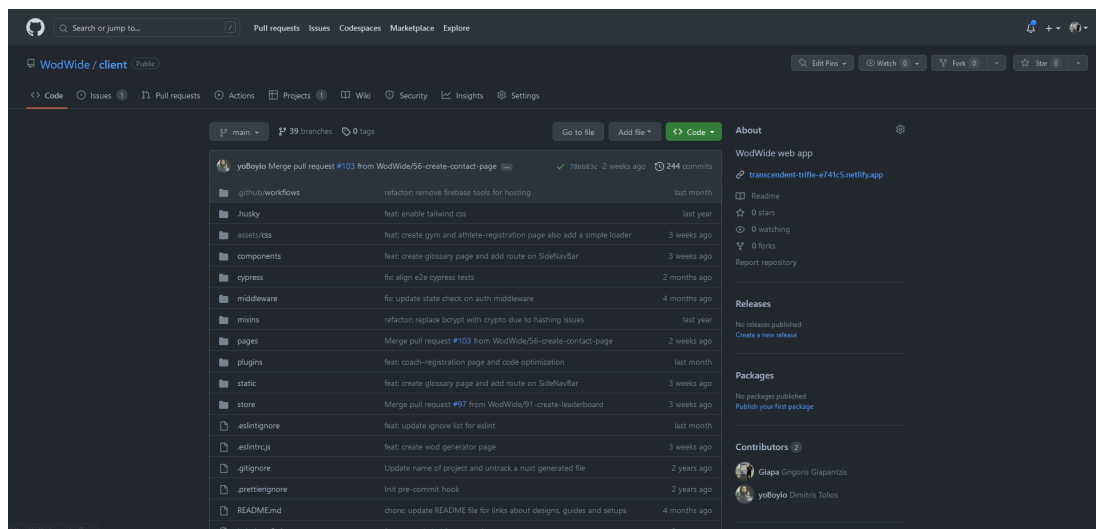


Figure 4.3: Github project

4.2.2 Github Actions

One of the key features of GitHub Actions is its flexibility in workflow automation. With GitHub Actions, developers can define workflows using YAML syntax, allowing for customizable and configurable automation frameworks. Workflows consist of one or more jobs, each containing a series of steps executed in response to specific events or triggers. This flexibility enables teams to automate repetitive tasks, such as building, testing, and deploying applications, reducing manual effort and increasing efficiency.

GitHub Actions boasts an extensive marketplace of pre-built actions, which are encapsulated units of work created by the community and GitHub. These actions cover a wide range of tasks, including running tests with popular testing frameworks, deploying applications to various cloud platforms, sending notifications, and much more. Leveraging these pre-built actions allows teams to tap into existing best practices and avoid reinventing the wheel, accelerating development processes and ensuring consistency.

Continuous integration and deployment (CI/CD) is another significant aspect of GitHub Actions. Teams can automatically trigger workflows based on events like code pushes, pull requests, or scheduled intervals. By automating the build and testing process, GitHub Actions helps catch issues early, ensuring code quality and reducing the risk of introducing bugs into the codebase. This enables teams to maintain a fast development pace while upholding high standards of quality and reliability.

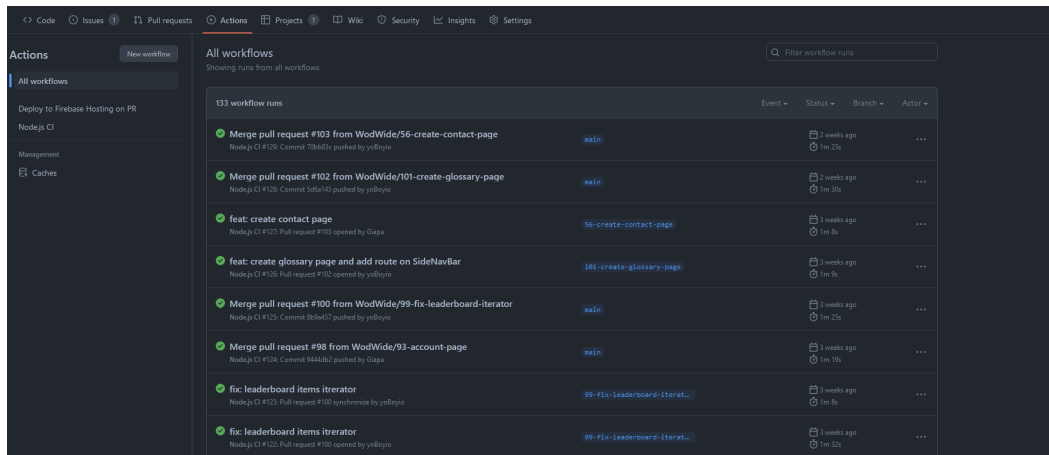


Figure 4.4: Github actions



Figure 4.5: Github actions YAML

4.2.3 Github Boards

GitHub Boards is a project management tool provided by GitHub that empowers teams to organize, track, and collaborate on their work effectively. With its intuitive interface and flexible features, GitHub Boards facilitates agile project management, enabling teams to plan, prioritize, and deliver their work in a collaborative and efficient manner.

One of the key features of GitHub Boards is its support for agile development methodologies. Agile methodologies, such as Scrum or Kanban, emphasize iterative and incremental development, fostering collaboration, adaptability, and continuous improvement. GitHub Boards aligns with these agile principles by providing customizable Kanban boards, allowing teams to visualize their tasks, track progress, and collaborate seamlessly.

GitHub Boards enables teams to create customizable boards that reflect their unique workflow and project structure. Boards consist of columns representing different stages of work, such as "To Do," "In Progress," and "Done." Team members can create project cards within each column to represent specific tasks, user stories, or issues. These cards can be assigned to team members, labeled, and categorized for easy organization and tracking.

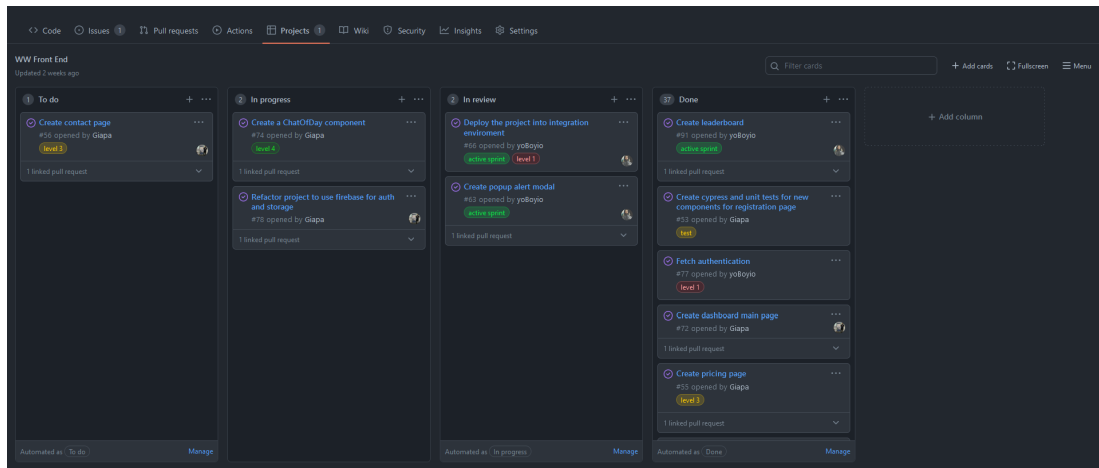


Figure 4.6: Github board

4.2.4 Github Issues

GitHub Issues provides a streamlined approach to issue tracking and bug reporting within a project. It allows team members to create, manage, and assign issues, facilitating effective communication and collaboration. With GitHub Issues, teams can track the progress of issue resolution, discuss potential solutions, and prioritize tasks. This tool enables efficient bug tracking, issue management, and contributes to the overall project quality.

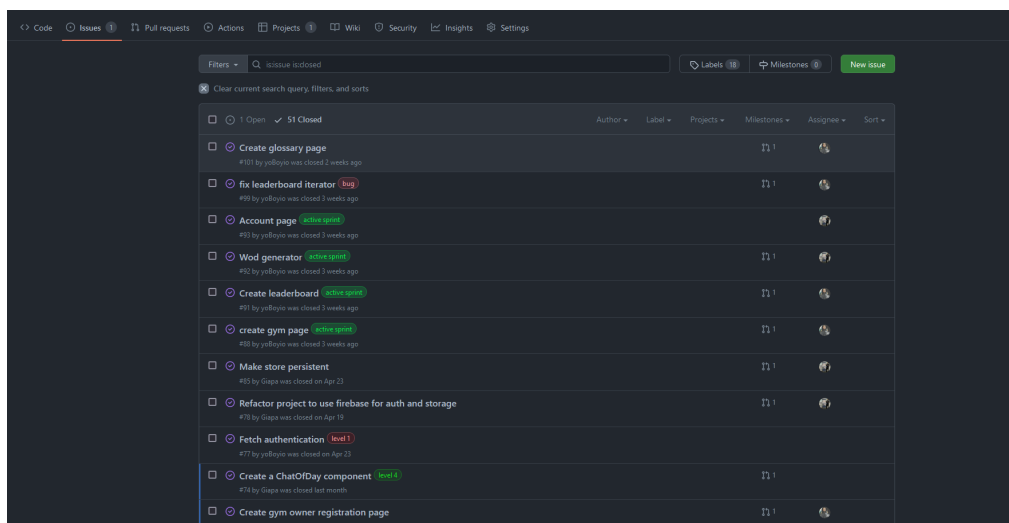


Figure 4.7: Github issues

4.3 Programming languages

4.3.1 Javascript

JavaScript has emerged as a fundamental programming language in modern web development, enabling dynamic and interactive experiences on the front end of websites and web applications. In this chapter, we explore the usage and significance of JavaScript in front-end development, highlighting its key features and capabilities that empower developers to create rich, engaging, and responsive user interfaces.

The Power of JavaScript in Front-End Development:

- **Dynamic Web Pages:** JavaScript's ability to manipulate and modify the Document Object Model (DOM) allows developers to create dynamic web pages that respond to user interactions. With JavaScript, developers can add or remove elements, update content, and apply visual effects on the fly. This dynamic nature enhances user engagement and provides a seamless and interactive browsing experience.
- **Event Handling:** JavaScript simplifies the process of handling user interactions and events on web pages. Developers can attach event listeners to various elements and respond to events such as button clicks, form submissions, mouse movements, and keyboard inputs. This event-driven paradigm enables the creation of responsive and interactive interfaces, allowing users to interact with the application in a natural and intuitive manner.
- **Data Manipulation and Validation:** JavaScript's extensive set of built-in functions and libraries empowers developers to manipulate, transform, and validate data on the client side. This capability reduces the reliance on server-side processing, improving performance and enhancing the user experience. JavaScript enables form validation, data formatting, and real-time data updates, ensuring data integrity and providing immediate feedback to users.
- **Asynchronous Programming:** JavaScript's support for asynchronous programming through features like Promises and asynchronous functions (async/await) facilitates the handling of time-consuming tasks, such as making network requests and fetching data from APIs. This non-blocking approach prevents the user interface from becoming unresponsive, allowing for smoother user experiences and efficient utilization of system resources.
- **Libraries and Frameworks:** JavaScript has a vast ecosystem of libraries and frameworks that simplify and accelerate front-end development. Popular libraries like React, Vue.js, and Angular provide powerful abstractions and component-based architectures, enabling developers to build scalable and maintainable user interfaces. These frameworks enhance productivity, facilitate code reusability, and promote best practices in front-end
- **Cross-Browser Compatibility:** JavaScript is supported by all modern web browsers, making it a universal language for front-end development. Developers can write code once and have it run seamlessly across different browsers and devices, ensuring consistent user experiences. Additionally, frameworks and libraries often include mechanisms to handle browser-specific quirks and ensure compatibility, further easing the development process.

4.3.2 Typescript

TypeScript, a superset of JavaScript, has gained significant popularity in recent years as a powerful language for front-end development. Building upon JavaScript's foundation, TypeScript brings additional advantages and tools that help developers write more robust, maintainable, and scalable code. In this chapter, we explore the unique features and benefits of TypeScript in front-end development, focusing on how it addresses common challenges and enhances the development process.

Advantages of TypeScript in Front-End Development:

- **Static Typing:** One of the key advantages of TypeScript is its support for static typing. By introducing type annotations, TypeScript enables developers to explicitly define variable types, func-

tion signatures, and interfaces. This helps catch type-related errors during development, improves code readability, and provides better tooling support, such as code autocompletion and refactoring. Static typing helps prevent runtime errors, reduces debugging time, and enhances overall code quality.

- **Enhanced Tooling and IDE Support:** TypeScript offers rich tooling and robust IDE support that significantly improves developer productivity. IDEs can leverage TypeScript's type information to provide intelligent code suggestions, error detection, and automatic refactoring capabilities. This level of tooling support allows developers to write code with greater confidence, navigate large codebases efficiently, and identify and fix issues early in the development process.
- **Better Maintainability and Scalability:** TypeScript's strong typing and additional language features, such as classes, interfaces, and modules, contribute to better code organization, maintainability, and scalability. With TypeScript, developers can define clear contracts between components, establish interfaces for data structures, and encapsulate logic within classes. These language constructs make codebases easier to understand, extend, and refactor, particularly in complex applications with multiple developers collaborating.
- **Code Documentation and Readability:** TypeScript promotes self-documenting code by providing a clear structure and strong typing. The use of type annotations serves as a form of documentation, making code more readable and understandable, both for developers and future maintainers. TypeScript's support for generating documentation through tools like TypeDoc further enhances code documentation, aiding in onboarding new team members and ensuring better code comprehension.
- **Safer Refactoring and Modularity:** Refactoring code is a common practice in software development, and TypeScript offers benefits in this area. With TypeScript's type system, renaming variables, changing function signatures, or modifying interfaces can be done safely, as the compiler helps identify and update all affected code references. This reduces the risk of introducing bugs during refactoring and allows for better code maintenance and evolution. TypeScript's support for modules and namespaces also facilitates code organization and modularity, making it easier to manage and reuse code across different parts of an application.

Closing the Gaps:

TypeScript addresses some of the limitations of JavaScript, such as its loose typing system, lack of compile-time checks, and limited tooling support. By introducing static typing and advanced language features, TypeScript helps close these gaps, providing developers with a more reliable and efficient development experience. It enables teams to build large-scale, complex front-end applications with confidence, while still benefiting from the vast ecosystem and compatibility with JavaScript.

4.3.3 Python

Python has established itself as a versatile and powerful programming language with widespread adoption in the field of backend development and machine learning. In this chapter, we explore the usage and significance of Python in these domains, highlighting its key features and capabilities that empower developers to build scalable and efficient backend systems and leverage machine learning algorithms to extract valuable insights from data.

Python in Backend Development:

- **Scalability and Performance:** Python's performance has significantly improved over the years, thanks to frameworks like Django and Flask optimizing their internals. Additionally, Python benefits from the ability to seamlessly integrate with high-performance languages like C and C++. This interoperability allows developers to write performance-critical sections of their backend code in lower-level languages, while leveraging Python for rapid prototyping, data processing, and glue code.
- **Database Integration:** Python offers excellent support for integrating with various databases, both SQL and NoSQL. Libraries like SQLAlchemy provide an Object-Relational Mapping (ORM) layer, enabling developers to interact with databases using Python objects and abstracting away the underlying database specifics. This simplifies database operations, enhances code maintainability, and promotes good practices in data modeling and query execution.
- **API Development:** Python excels in building robust APIs (Application Programming Interfaces) that allow different systems to communicate and exchange data. Libraries such as FastAPI and Flask-RESTful make it straightforward to develop RESTful APIs, handle request parsing, authentication, serialization, and validation. Python's simplicity and readability make it an ideal choice for building APIs that are easy to understand, maintain, and consume.

Python in Machine Learning:

- **Rich Ecosystem of Libraries:** Python's dominance in machine learning stems from its extensive ecosystem of libraries and frameworks. Libraries like NumPy, pandas, and scikit-learn provide powerful tools for data manipulation, analysis, and feature engineering. TensorFlow and PyTorch offer high-level APIs for building and training neural networks, while libraries like Keras provide a user-friendly interface for rapid prototyping and experimentation.
- **Data Processing and Visualization:** Python's simplicity and expressive syntax make it an excellent choice for data processing and visualization tasks in machine learning. Libraries like pandas and NumPy enable efficient data manipulation, cleaning, and preprocessing. Matplotlib and Seaborn provide rich visualization capabilities, allowing developers to create informative plots and charts to gain insights from data and present results effectively.
- **Easy Prototyping and Experimentation:** Python's dynamic nature and interactive development environment make it an ideal language for rapid prototyping and experimentation in machine learning. Jupyter Notebook and JupyterLab provide an interactive coding environment where developers can easily experiment, document, and visualize their machine learning models. This iterative process accelerates development cycles, encourages exploration, and facilitates collaboration in the field of machine learning.
- **Integration with Deep Learning:** Python's popularity in machine learning is further enhanced by its seamless integration with deep learning frameworks like TensorFlow and PyTorch. These frameworks offer extensive support for building and training deep neural networks, allowing developers to leverage the power of deep learning architectures for complex tasks such as image recognition, natural language processing, and reinforcement learning.

4.4 Nuxt

Nuxt.js, a framework built on top of Vue.js, has gained popularity as a powerful tool for developing server-side rendered (SSR) applications with excellent search engine optimization (SEO) capabilities. In this chapter, we explore the purpose and advantages of Nuxt.js, focusing on how it enables developers to build SEO-friendly applications through server-side rendering.

Advantages of Nuxt.js:

- **SEO-friendly Server-Side Rendering (SSR):** Nuxt.js excels in providing SSR capabilities, which have a significant impact on search engine optimization. SSR allows search engines to crawl and index the fully rendered HTML content of web pages. By pre-rendering pages on the server, Nuxt.js ensures that search engine bots receive complete HTML responses, leading to improved discoverability and ranking in search engine results. SSR also enhances initial page load performance, resulting in a better user experience.
- **Automatic Routing and Code Splitting:** Nuxt.js simplifies the implementation of client-side routing through its built-in routing system. By following conventions, Nuxt.js automatically generates routes based on the file structure, eliminating the need for manual route configuration. Additionally, Nuxt.js implements code splitting, which optimizes the loading of JavaScript by splitting it into smaller chunks. This reduces the initial load time and improves overall performance, both of which positively impact SEO.
- **Vue.js Ecosystem and Vue Router Integration:** Nuxt.js benefits from its integration with the Vue.js ecosystem, leveraging the power and flexibility of Vue components and the Vue Router. Developers can create reusable and modular components using Vue.js, promoting code reusability and maintainability. The Vue Router integration in Nuxt.js simplifies the management of routing, including dynamic routes and nested routes, enabling developers to build complex applications with ease.
- **Middleware and Server Middleware:** Nuxt.js provides middleware functionality that allows developers to handle specific tasks before rendering a page. Middleware can be used for tasks such as authentication, data fetching, and redirecting, providing flexibility in handling common server-side operations. Server middleware extends this functionality by enabling developers to add custom server-side logic to the application, enhancing the server-rendering process and opening up possibilities for advanced server-side operations.
- **Nuxt.js Modules:** Nuxt.js introduces the concept of modules, which are reusable units of functionality that can be easily added to the project. Modules provide pre-configured features such as state management, authentication, content management systems integration, and more. These modules simplify the integration of common functionality into Nuxt.js applications, reducing development time and effort.

SEO Benefits of Nuxt.js SSR:

- **Improved Indexing:** By pre-rendering pages on the server, Nuxt.js ensures that search engine bots receive fully rendered HTML content, facilitating better indexing and discoverability of web pages. This leads to improved search engine rankings and visibility for websites built with Nuxt.js.

- **Faster Initial Page Load:** SSR provided by Nuxt.js significantly improves the initial page load performance, resulting in better user experiences. Fast-loading pages are not only appreciated by users but also favored by search engines, as they contribute to a positive user experience.
- **Social Media Sharing and Link Previews:** Nuxt.js SSR enables proper rendering of web pages when shared on social media platforms. Social media crawlers receive complete HTML content, allowing for accurate link previews and better presentation of shared links, ultimately enhancing the visibility and click-through rates on social media.

4.5 Styling

Effective website styling plays a crucial role in creating visually appealing and user-friendly web experiences. In this chapter, we delve into the fundamentals of styling a website, exploring key technologies such as CSS, Tailwind CSS, and SCSS. We'll discuss how these tools empower developers to create stunning designs, improve maintainability, and enhance the efficiency of the styling process.

4.5.1 Tailwing

Tailwind CSS has gained significant popularity as a utility-first CSS framework that simplifies and accelerates the web development process. In this chapter, we explore the key features and advantages of Tailwind CSS, highlighting how it streamlines styling workflows, promotes consistency, and empowers developers to create highly customizable and responsive designs.

What is Tailwind CSS?

Tailwind CSS is a highly configurable CSS framework that focuses on providing an extensive collection of utility classes. Unlike traditional CSS frameworks that rely on pre-designed components, Tailwind CSS offers a utility-first approach. It provides a vast set of utility classes that developers can apply directly to HTML elements to achieve specific styling and layout effects. Tailwind CSS embraces a "low-level" approach, enabling granular control over every aspect of styling without sacrificing flexibility.

Key Features and Benefits:

- **Utility-First Approach:** Tailwind CSS revolves around utility classes that represent individual CSS properties. These classes are designed to handle common styling tasks, such as margins, padding, colors, typography, and responsive layouts. By composing these utility classes, developers can quickly assemble complex layouts and style elements, reducing the need for writing custom CSS code. This approach promotes consistency and accelerates development by eliminating the need to redefine styles repeatedly.
- **Highly Configurable:** Tailwind CSS offers extensive configuration options, allowing developers to customize the framework according to project requirements. From color palettes to spacing scales, font sizes to breakpoints, Tailwind CSS provides a configuration file that enables developers to define their own design system. This flexibility ensures that Tailwind CSS can adapt to various design languages and branding guidelines.

- **Responsive Design Made Easy:** Tailwind CSS simplifies the creation of responsive designs by providing intuitive breakpoint utilities. Developers can apply responsive classes to control the appearance of elements at different screen sizes, making it effortless to build mobile-friendly and responsive websites. The built-in responsive design utilities eliminate the need for media queries and enable developers to focus on the core layout and design logic.
- **Extensive Component Library:** Tailwind CSS has a comprehensive library of pre-built components called "component styles." These styles are reusable utility class combinations that encapsulate common design patterns. The component styles library provides a starting point for developers, allowing them to assemble complex components quickly and maintain consistency across the application. Developers can also create custom component styles based on their project's specific requirements.
- **Developer Experience and Tooling:** Tailwind CSS offers excellent developer experience and tooling support. It integrates seamlessly with popular build tools like webpack and PostCSS, enabling efficient compilation and optimization of stylesheets. Tailwind CSS also provides a powerful utility class editor called "Tailwind UI," which offers a visual interface for exploring and customizing the utility classes, facilitating a rapid iteration process.

```
<div class="flex flex-col p-4">
  <div
    class="gym-page__title flex items-center justify-center font-lg text-lg lg:text-3xl self-center w-fit shadow-xl"
    :class="{ 'dashboard-page__title--mobile': isMobile }"
  >
    Gym Page
    
  </div>
  <div v-if="loading" class="self-center mt-20">
    <font-awesome-icon class="text-2xl" :icon="['fas', 'spinner']" spin />
  </div>
```

Figure 4.8: Tailwind inside a Nuxt component

4.5.2 SCSS

SCSS (Syntactically Awesome Style Sheets) is a powerful preprocessor for CSS that enhances the styling capabilities of web developers. In this chapter, we explore the features and benefits of SCSS, highlighting how it improves code organization, promotes reusability, and empowers developers to create maintainable and efficient stylesheets.

What is SCSS?

SCSS is a superset of CSS that introduces advanced features, including variables, mixins, nesting, and more, while maintaining compatibility with the CSS syntax. SCSS files are preprocessed to generate CSS stylesheets, allowing developers to write CSS in a more concise and efficient manner. By leveraging the features offered by SCSS, developers can enhance the flexibility, modularity, and maintainability of their stylesheets. [12]

Key Features and Benefits:

- **Variables:** One of the key features of SCSS is the ability to define and use variables. Developers can declare variables to store values such as colors, font sizes, spacing, or any other design-related properties. By using variables, developers can ensure consistency throughout the stylesheet, making it easier to update and maintain styles. Variables also enable developers to reuse values and reduce redundancy, resulting in more efficient and DRY (Don't Repeat Yourself) code.
- **Mixins:** SCSS provides mixins, which are reusable blocks of styles that can be included in multiple selectors. Mixins allow developers to define complex styles once and then apply them to various elements. This promotes code reusability, reduces duplication, and simplifies the maintenance of styles across different parts of a project. Mixins can accept arguments, making them even more flexible and adaptable to different use cases.
- **Partials and Imports:** SCSS supports partials, which are files containing snippets of SCSS code. Partials are typically used to break down stylesheets into smaller, manageable modules. They can be imported into other SCSS files, allowing for modular organization and separation of concerns. This modular approach makes it easier to collaborate with other developers and maintain a scalable codebase.
- **Partials and Imports:** SCSS supports partials, which are files containing snippets of SCSS code. Partials are typically used to break down stylesheets into smaller, manageable modules. They can be imported into other SCSS files, allowing for modular organization and separation of concerns. This modular approach makes it easier to collaborate with other developers and maintain a scalable codebase.

```
<style lang="scss">
.gym-page {
  &__title {
    z-index: 10;
    background: white;
    padding: 1em;
    border-radius: 1em;
    &--mobile {
      left: 27%;
      width: 60%;
      top: 4em;
      font-size: 0.75em;
    }
  }
}
&__content {
  @apply self-center;
  margin-top: 4em;
  padding: 4em;
  border: 2px solid black;
  border-radius: 2em;
}
&__details {
  margin-top: 4em;
  margin-left: 7em;
  padding: 2em;
  border: 2px solid black;
  border-radius: 2em;
}
}
</style>
```

Figure 4.9: Scss inside a Nuxt component

4.6 Firebase

Firebase, developed by Google, is a comprehensive platform that provides a wide range of cloud-based services and tools for building and scaling web and mobile applications. In this chapter, we explore the features and benefits of Firebase, highlighting how it simplifies backend development, accelerates app development cycles, and enables developers to focus on creating engaging user experiences.

What is Firebase?

Firebase is a Backend-as-a-Service (BaaS) platform that offers a suite of tools and services to handle various aspects of modern application development. It provides developers with ready-to-use functionality for authentication, real-time databases, cloud storage, hosting, machine learning, analytics, and more. Firebase's powerful set of tools enables developers to build high-quality applications without having to manage complex server infrastructure or invest significant resources in backend development. [13]

Key Features and Benefits:

- **Real-Time Database:** Firebase's real-time database is a NoSQL database that allows developers to store and sync data in real-time across multiple clients. It enables real-time collaboration, making it ideal for applications that require instant updates, such as chat apps, collaborative editing tools, or real-time dashboards. The real-time database offers offline capabilities, ensuring that users can continue to work even when there's no internet connection. It also helps reduce redundancy, resulting in more efficient and DRY (Don't Repeat Yourself) code.
- **Authentication:** Firebase provides a robust authentication system that simplifies user management and authentication processes. It supports various authentication methods, including email/password, social logins (such as Google, Facebook, and Twitter), and third-party identity providers. With Firebase Authentication, developers can easily implement secure user authentication and authorization functionalities without the need for complex backend infrastructure.
- **Cloud Firestore:** Cloud Firestore is a flexible and scalable NoSQL document database offered by Firebase. It provides a powerful data storage solution that supports automatic data syncing, offline capabilities, real-time updates, and robust querying capabilities. Cloud Firestore seamlessly integrates with other Firebase services, allowing developers to build data-driven applications efficiently.
- **Hosting and Deployment:** Firebase Hosting simplifies the deployment and hosting process for web applications. It offers a global content delivery network (CDN) that ensures fast and reliable content delivery to users worldwide. Firebase Hosting supports single-page applications, dynamic content, custom domains, HTTPS encryption, and automatic scaling. It eliminates the need for managing server infrastructure and simplifies the process of deploying web applications.

4.7 FastAPI

FastAPI is a modern, high-performance web framework for building APIs with Python. In this chapter, we delve into the features and advantages of FastAPI, highlighting how it enables developers to create efficient, scalable, and secure backend APIs. With its emphasis on speed, type hints, and asynchronous

support, FastAPI has gained popularity among developers seeking a productive and robust framework for API development.

What is FastAPI?

FastAPI is a Python web framework that is designed to be easy to use, highly efficient, and scalable. It leverages Python 3.7+ type hints and asynchronous programming to provide a performant and intuitive API development experience. FastAPI is built on top of the Starlette framework, which offers powerful features for handling HTTP requests and responses. It also integrates with popular Python tools and libraries, making it a versatile choice for building APIs. [14]

Key Features and Benefits:

- **Fast and Performant:** FastAPI is designed to deliver high-performance web APIs. By utilizing asynchronous programming and the power of Starlette, FastAPI can handle a large number of concurrent requests efficiently. It leverages the performance benefits of Python's asynchronous capabilities, allowing for non-blocking operations and efficient resource utilization. FastAPI's performance makes it suitable for applications that require real-time processing, high traffic, or low latency.
- **Modern Python Features:** FastAPI takes advantage of modern Python features to enhance developer productivity and code reliability. It utilizes type annotations to provide automatic data validation and serialization, reducing the chances of runtime errors. FastAPI also leverages Python's asynchronous capabilities to handle multiple requests concurrently, resulting in improved efficiency and responsiveness.
- **Interactive Documentation:** FastAPI automatically generates interactive documentation for APIs based on the provided type annotations and docstrings. The documentation includes detailed information about endpoints, request/response models, data types, and available operations. The generated documentation supports features like request parameter validation, input form generation, and interactive API exploration, making it easy for developers to understand and consume the API.
- **Built-in Validation and Serialization:** FastAPI integrates seamlessly with Pydantic, a powerful library for data validation and serialization. By leveraging Pydantic's features, FastAPI can automatically validate request data against defined models, handle type conversions, and provide detailed error messages. This simplifies the validation and serialization process, reducing the boilerplate code required for data processing and improving the overall robustness of the API.
- **Extensible and Customizable:** FastAPI allows developers to extend its capabilities through middleware, dependency injection, and custom components. It supports the use of middleware functions to modify request/response handling, enabling developers to add custom logic at various stages of the API pipeline. FastAPI's dependency injection system facilitates clean and maintainable code by managing component dependencies automatically. Additionally, developers can customize FastAPI's behavior by overriding default settings and configurations to suit their project requirements.

4.8 Testing tools

Testing tools play a crucial role in ensuring that software applications meet functional requirements, perform as expected, and remain bug-free. They provide a systematic approach to verifying the behavior of code, identifying potential issues, and validating the application's functionality across various scenarios. Effective testing tools offer features such as test automation, assertion libraries, mocking capabilities, and extensive reporting, enabling developers to catch bugs early, improve code quality, and deliver reliable software to end-users.

4.8.1 Nuxt-testing-utils

Nuxt Testing Utils is a dedicated testing utility library designed to streamline the testing process for Nuxt.js applications. Nuxt.js, built on top of Vue.js, is a powerful framework for creating server-rendered and static websites. Nuxt Testing Utils provides a range of utilities and assertions specifically tailored for Nuxt.js applications, making it easier for developers to write comprehensive and reliable tests. [15]

Key Features and Benefits:

- **Simplified Testing Setup:** Nuxt Testing Utils simplifies the setup process for testing Nuxt.js applications. It provides a test harness that automatically configures the necessary environment for testing Nuxt components, pages, and Vuex store modules. This includes setting up a Vue instance with a mocked Nuxt context, Vuex store, and router, allowing developers to focus on writing test cases rather than dealing with complex setup procedures.
- **Component and Page Testing:** With Nuxt Testing Utils, developers can easily write unit tests for Nuxt components and pages. The library offers a variety of helper functions and assertions specifically designed for testing Nuxt components, such as `shallowMount`, `mount`, and `findComponent`. These functions facilitate the creation of isolated component tests, enabling developers to assert the expected behavior of individual components and their interactions with props, data, computed properties, and methods.
- **Mocking API Requests:** Nuxt Testing Utils provides utilities for mocking API requests in Nuxt.js applications. It allows developers to intercept and mock HTTP requests made by the application, simulating different server responses. This capability is particularly useful when testing components or pages that rely on asynchronous data fetching or API interactions. By mocking API requests, developers can control the responses and ensure consistent behavior during testing.
- **Vuex Store Testing:** Nuxt Testing Utils offers built-in support for testing Vuex store modules. Developers can easily mock and test Vuex actions, mutations, and getters using the provided utilities. The library allows for easy manipulation of the store's state and dispatching actions, enabling developers to thoroughly test the behavior of their Vuex store modules in different scenarios.
- **Integration with Testing Frameworks:** Nuxt Testing Utils seamlessly integrates with popular JavaScript testing frameworks such as Jest and Vue Test Utils. It provides additional functionalities and helpers specifically designed for testing Nuxt.js applications, making it easier to incorporate testing into existing development workflows and testing setups.

- **Test Coverage and Reliability:** By using Nuxt Testing Utils to write tests for Nuxt.js applications, developers can achieve better test coverage and ensure the reliability of their code. The utilities and assertions provided by the library enable developers to cover a wide range of scenarios, making it easier to detect and fix issues before they reach the production environment. This, in turn, improves code quality, reduces the risk of bugs, and enhances the overall stability of Nuxt.js applications.

4.8.2 Cypress

Cypress is a robust end-to-end testing framework that provides a comprehensive solution for testing web applications. It offers a range of features and capabilities specifically designed to simplify and streamline the process of testing web applications, ensuring their functionality, performance, and reliability. [?]

Key Features and Benefits:

- **Real-time Interactive Testing:** Cypress offers a unique real-time interactive testing experience. Developers can view the application being tested in real-time as the tests are executed, allowing them to observe the application's behavior and interact with it during the testing process. This real-time view provides immediate feedback and enhances the debugging process, making it easier to identify and resolve issues.
- **Automatic Waiting and Retries:** Cypress automatically waits for elements and actions to be available and interactable before proceeding with the test. This eliminates the need for manual timeouts and delays, resulting in more stable and reliable tests. Additionally, Cypress automatically retries failed assertions, improving the robustness of test execution.
- **Time-Travel Debugging:** One of the standout features of Cypress is its time-travel debugging capability. Developers can inspect and debug their application's state at any point during test execution. This allows for efficient troubleshooting and pinpointing the root cause of failures. By stepping through the application's state at different stages of the test, developers gain valuable insights into the behavior and interactions of the application.
- **Easy Installation and Setup:** Cypress has a simple and straightforward installation process. It is distributed as a single package and can be installed with just a few commands. Cypress also provides a user-friendly graphical interface that allows developers to easily configure and manage their test suite, select test files to run, and view test results and reports.
- **Powerful and Expressive API:** Cypress provides a powerful and expressive API that simplifies the process of writing tests. The API allows developers to interact with the application under test, perform assertions, and simulate user actions such as clicks, form submissions, and keyboard input. This enables developers to write clean and readable test cases, making it easier to understand and maintain the test suite.
- **End-to-End Testing Capabilities:** Cypress supports end-to-end testing, allowing developers to test the entire application stack, including UI interactions, network requests, and data flows. It can simulate user behavior across different pages and routes, making it possible to test complex user flows and scenarios. Cypress also offers network stubbing and mocking capabilities, allowing developers to control and intercept network requests during testing.

- **Extensive Plugin Ecosystem:** Cypress has a vibrant and active plugin ecosystem that extends its capabilities. Developers can leverage a wide range of plugins to enhance their testing workflow, integrate with third-party tools and services, and extend Cypress's functionality to meet specific testing requirements.

4.9 Netlify

Netlify is a powerful web hosting and continuous deployment platform that simplifies the process of deploying and managing websites and web applications. In this chapter, we explore the features and benefits of Netlify, highlighting how it streamlines the deployment workflow, enables seamless continuous integration and deployment, and provides a scalable and reliable hosting infrastructure. [16]

Key Features and Benefits:

- **Continuous Deployment:** Netlify offers seamless continuous deployment capabilities, enabling developers to automate the process of deploying websites and web applications. With its Git-based workflow, Netlify automatically builds and deploys applications whenever changes are pushed to a connected Git repository. This eliminates the need for manual deployment processes, saving time and reducing the risk of human error.
- **Git Integration:** Netlify seamlessly integrates with popular Git repositories, such as GitHub and Bitbucket. This integration allows developers to connect their repositories to Netlify and trigger automatic builds and deployments based on changes made to the codebase. The tight integration with Git repositories simplifies the deployment process and ensures that the latest version of the application is always deployed.
- **Scalable and Reliable Hosting:** Netlify provides a scalable and reliable hosting infrastructure for websites and web applications. It leverages a global Content Delivery Network (CDN) to deliver content quickly and efficiently to users around the world. Netlify automatically optimizes assets, including images and JavaScript files, and ensures that the website is served from the closest server location to the end-users, resulting in faster loading times and improved performance.
- **Instant Rollbacks and Versioning:** Netlify offers versioning and rollback capabilities, allowing developers to easily roll back to previous versions of their deployed applications. This is particularly useful in situations where issues arise after a deployment. With Netlify's versioning, developers can easily revert to a known working state of the application, minimizing downtime and ensuring a seamless user experience.
- **Built-in Form Handling and Backend Functionality:** Netlify provides built-in form handling capabilities, allowing developers to capture and process form submissions without the need for server-side code. Netlify can receive form submissions, store them securely, and send notifications or trigger workflows based on the received data. This eliminates the need for a separate backend server for form processing, simplifying the architecture and reducing maintenance overhead.
- **Analytics and A/B Testing:** Netlify offers built-in analytics and A/B testing capabilities, providing insights into website performance, visitor behavior, and conversion rates. Developers can track key metrics, set up A/B tests, and make data-driven decisions to optimize the user experience and

improve business outcomes. The built-in analytics and A/B testing features eliminate the need for third-party integrations, simplifying the monitoring and optimization process.

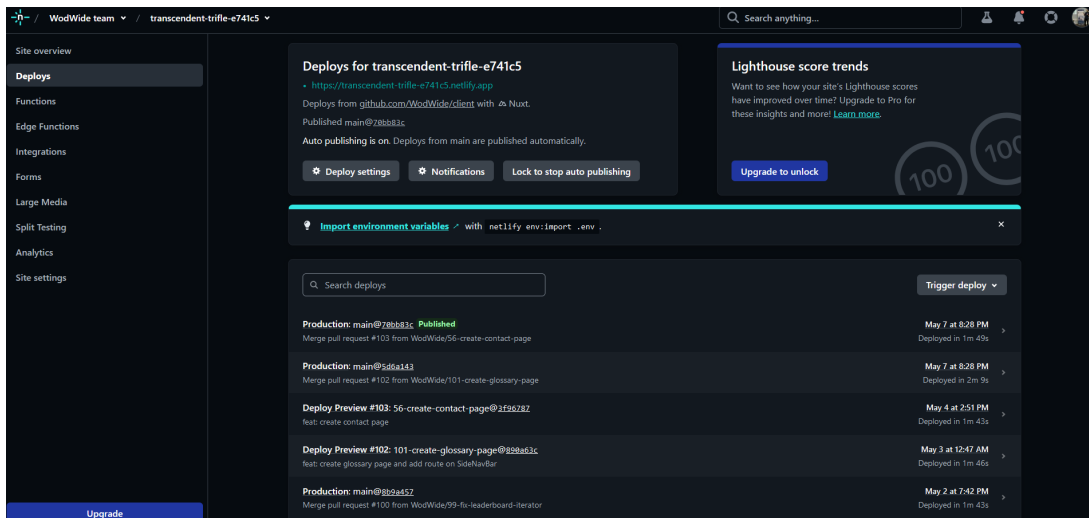


Figure 4.10: WodWide panel on Netlify

Chapter 5: WodWide Design

5.1 WodWide architecture - UML

WodWide is a web application designed to cater to the needs of CrossFit enthusiasts. It provides users with a platform to explore and engage in various CrossFit-related activities, including accessing workout leaderboards, joining gyms, generating custom workouts, and more. The architecture of WodWide is designed to provide a seamless user experience while maintaining the necessary functionalities and data flow.

The presentation layer handles user interactions and displays the web pages to the clients. It receives user inputs and triggers corresponding actions, such as visiting the home page, about page, pricing page, contact page, or accessing the dashboard. Additionally, it allows users to perform actions like sending emails, logging in, signing up, viewing leaderboards, joining gyms, generating custom workouts, managing account information, exploring the glossary, adjusting settings, and logging out.

The business logic layer contains the core functionalities of the application. It processes the user requests received from the presentation layer and coordinates the necessary operations. It handles tasks such as retrieving and updating data, generating custom workouts based on user preferences and equipment, managing user accounts, handling email communication, and providing access to gym information.

The data access layer is responsible for interacting with the underlying database or any other persistence mechanism used by WodWide. It facilitates the storage and retrieval of user-related data, workout information, leaderboard data, and other necessary resources.

Overall, the architecture of WodWide is designed to provide users with a seamless and engaging CrossFit experience. By leveraging a client-server model and employing layered architecture, the application offers a user-friendly interface, efficient data processing, and robust functionalities that cater to the needs of CrossFit enthusiasts.

To illustrate the structure and interactions within WodWide, the following UML diagrams have been created. These diagrams depict the different actors involved, the various pages and functionalities available to users, and the relationships between them. The diagrams provide a visual representation of how users can navigate through the application, access different features, and interact with the system.

It is important to note that the UML diagrams provided here offer a high-level overview of the WodWide web application architecture. The diagrams focus on the user interactions, showcasing the main pages and actions available to them. However, the underlying implementation details, technologies, and specific software components may vary based on the actual development and deployment of the application.

Please refer to the accompanying UML diagrams for a visual representation of the WodWide web application architecture, detailing the user actions and the corresponding system behavior. These diagrams will help in understanding the flow and functionality of the application, aiding in the development and analysis of the WodWide system.

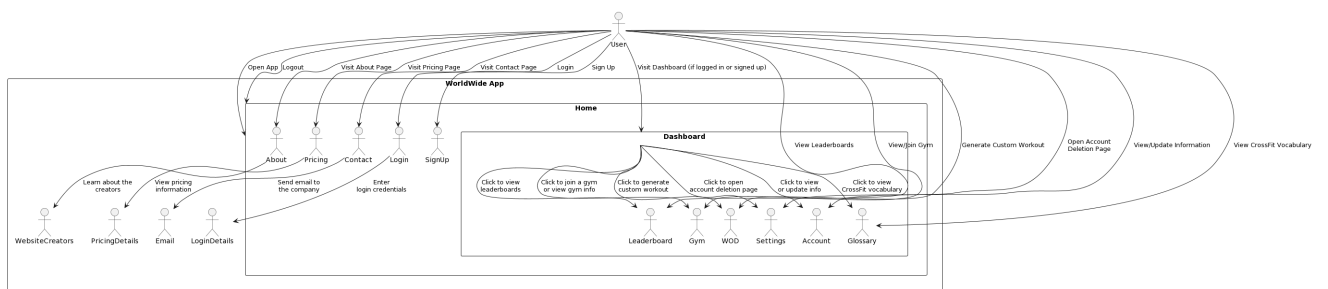


Figure 5.1: UML representation

```

"Worldwide App" {
  User --> Home : Open App

  Home {
    User --> About : Visit About Page
    User --> Pricing : Visit Pricing Page
    User --> Contact : Visit Contact Page
    User --> Login : Login
    User --> SignUp : Sign Up
    User --> Dashboard : Visit Dashboard (if logged in or signed up)
  }

  About --> WebsiteCreators : Learn about the creators
  Pricing --> PricingDetails : View pricing information
  Contact --> Email : Send email to the company
  Login --> LoginDetails : Enter login credentials

  Dashboard {
    User --> Leaderboard : View Leaderboards
    User --> Gym : View/Join Gym
    User --> WOD : Generate Custom Workout
    User --> Account : View/Update Information
    User --> Glossary : View CrossFit Vocabulary
    User --> Settings : Open Account Deletion Page
    User --> Home : Logout
  }

  Dashboard --> Leaderboard : Click to view leaderboards
  Dashboard --> Gym : Click to join a gym or view gym info
  Dashboard --> WOD : Click to generate custom workout
  Dashboard --> Account : Click to view or update info
  Dashboard --> Glossary : Click to view CrossFit vocabulary
  Dashboard --> Settings : Click to open account deletion page
}

```

Figure 5.2: UML text representation

5.2 Backend

The backend architecture of the WodWide app incorporates Firebase for user authentication, data storage, and real-time chat functionality, along with FastAPI for handling requests and integrating with the machine learning (ML) model. Let's explore the components and their interactions:

- **Firebase Authentication:** Firebase Authentication is utilized for user authentication and account creation. It provides a secure and scalable solution for managing user authentication, including options for email/password authentication, social media logins, and more. Users can sign up, log in, and authenticate themselves securely, gaining access to the app's features and functionalities.
- **Firebase Firestore:** Firebase Firestore is employed as the database for storing information about athletes, users, and gyms. Firestore is a NoSQL document-based database that offers real-time synchronization and scalability. It allows for efficient storage and retrieval of structured data, enabling the app to store and manage athlete profiles, user data, and gym-related information. Firestore's real-time capabilities ensure that the data stays synchronized across different devices and provides seamless access to up-to-date information.
- **Firebase Realtime Database:** Firebase Realtime Database is utilized for creating chat functionality within the app. It enables athletes to communicate with other gym members through real-time messaging. The Realtime Database offers a NoSQL JSON-based data model, allowing for efficient and synchronized updates across multiple clients. Athletes can send and receive messages in real-time, facilitating communication and community-building within the app.
- **FastAPI:** FastAPI serves as the backend framework for receiving requests from the frontend and interacting with the ML model. FastAPI is a modern, high-performance Python web framework that allows for efficient handling of HTTP requests. It provides a straightforward interface for defining API endpoints, handling input validation, and processing requests asynchronously. FastAPI acts as a bridge between the frontend and the ML model, receiving parameters from the frontend, passing them to the ML model, and returning the results to the frontend.

Overall, this backend architecture leverages Firebase for user authentication, data storage, and real-time chat functionality. FastAPI acts as the web framework to handle requests from the frontend and interact with the ML model, enabling seamless communication between the frontend, backend, and ML components of the WodWide app.

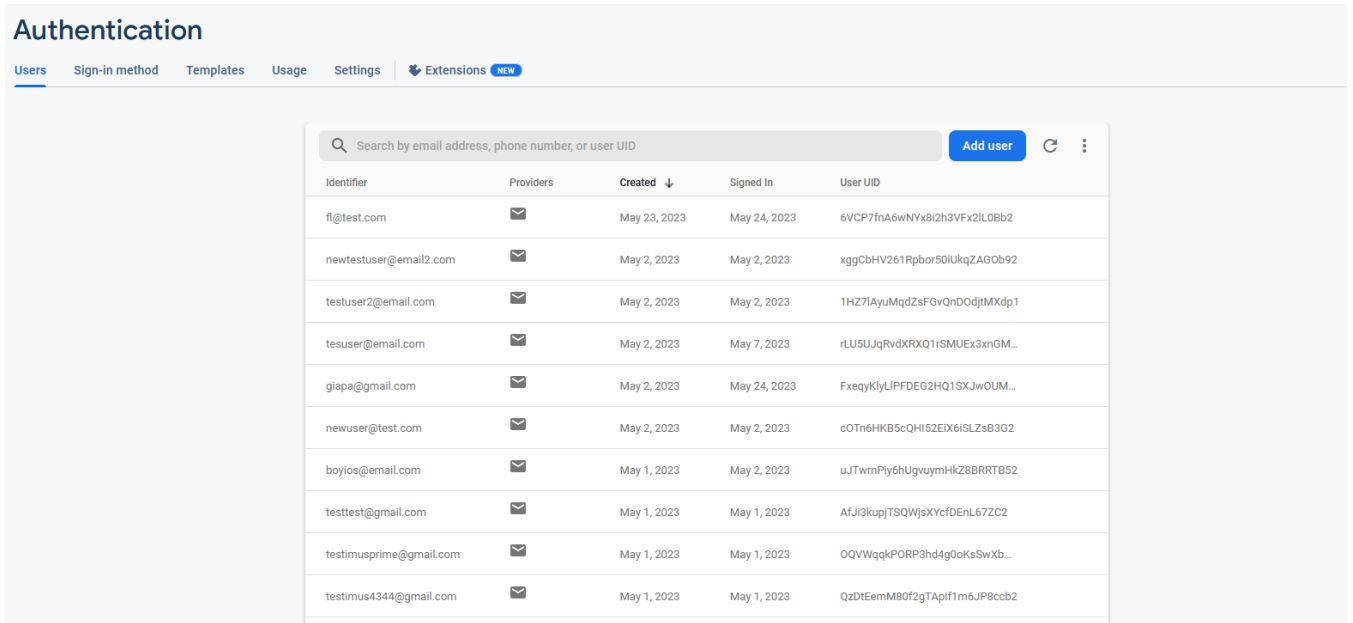


Figure 5.3: Firebase dashboard for Authentication

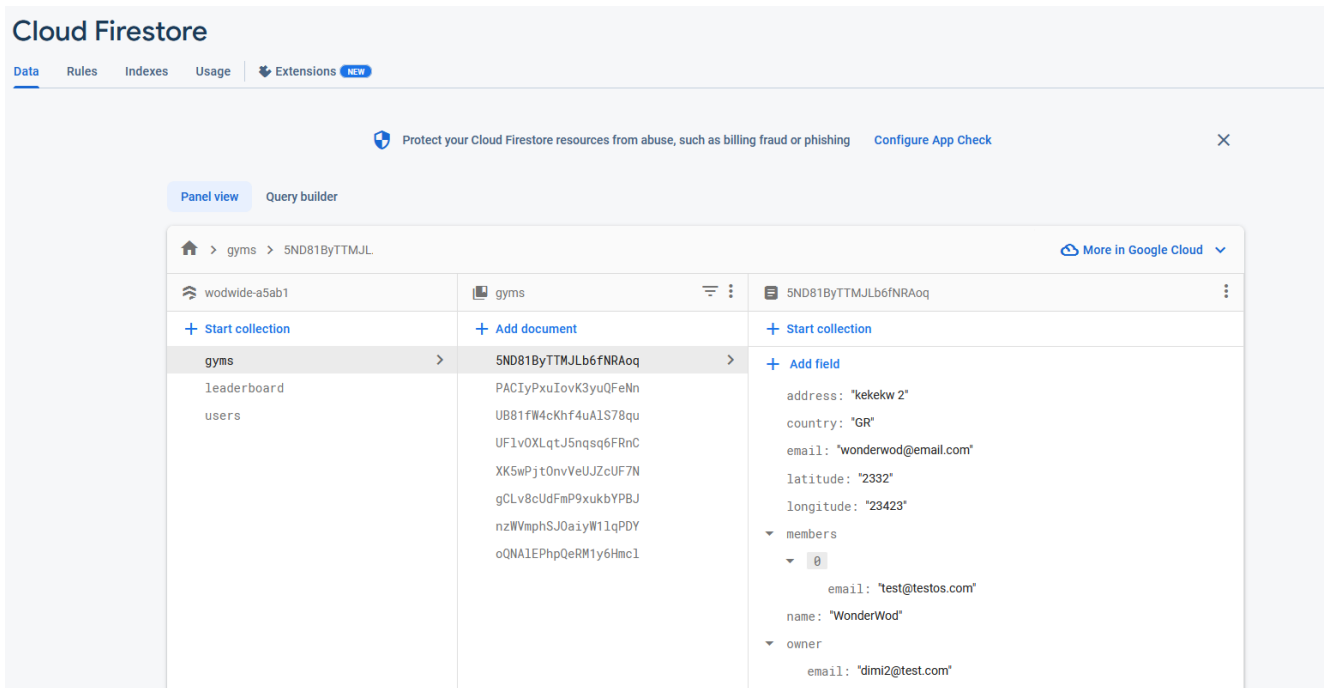


Figure 5.4: Firebase dashboard for Data Storage

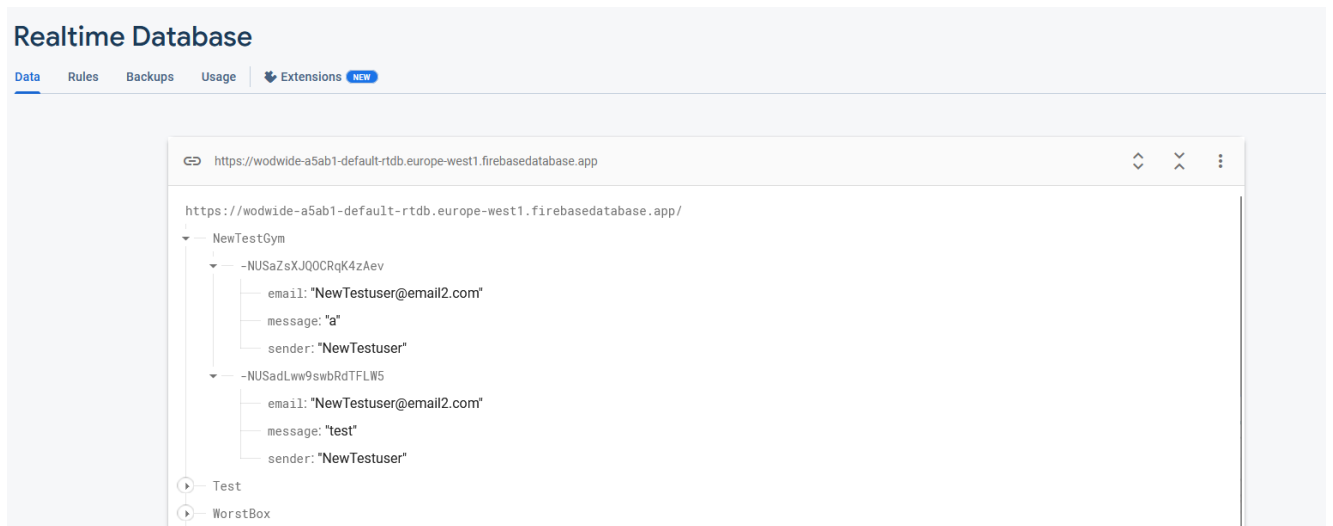


Figure 5.5: Firebase dashboard for Realtime Database

5.3 Frontend

Chapter 6: WodWide presentation

This section will showcase the application and explore all potential use cases, encompassing both athlete and coach perspectives. On the Home page[Figure 6.1], users have the option to click on a button that will take them to the Login page if they already have an account, or to the Signup page if they don't. Additionally, there are buttons available to navigate to the About page and the Contact page. Users can utilize the Contact page[Figure 6.2] to get in touch with us for any inquiries or information they may need. The About page[Figure 6.3] contains information about us and our responsibilities. On the Login page[Figure 6.4], users have the ability to access the app by logging in using their existing account credentials.

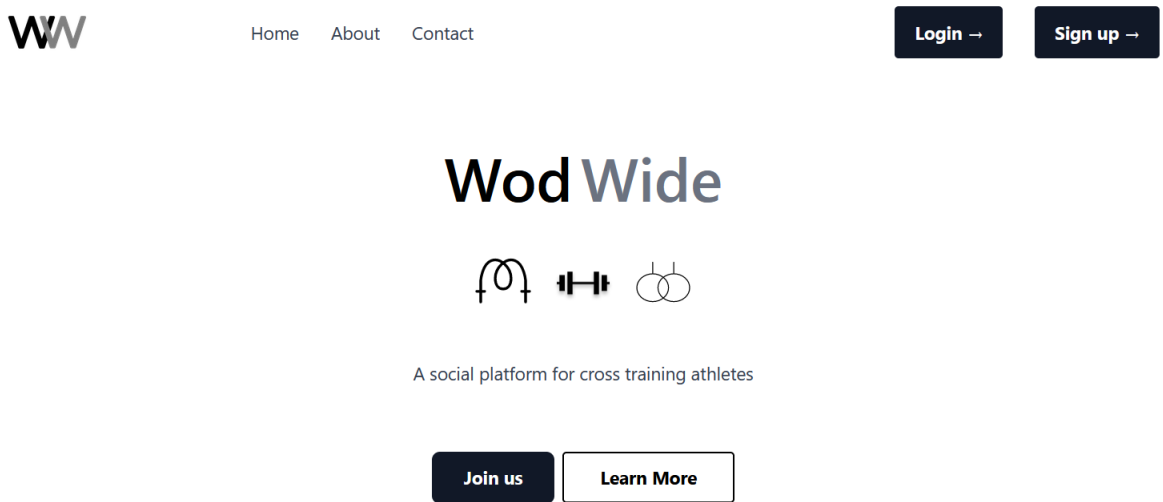


Figure 6.1: Home Page

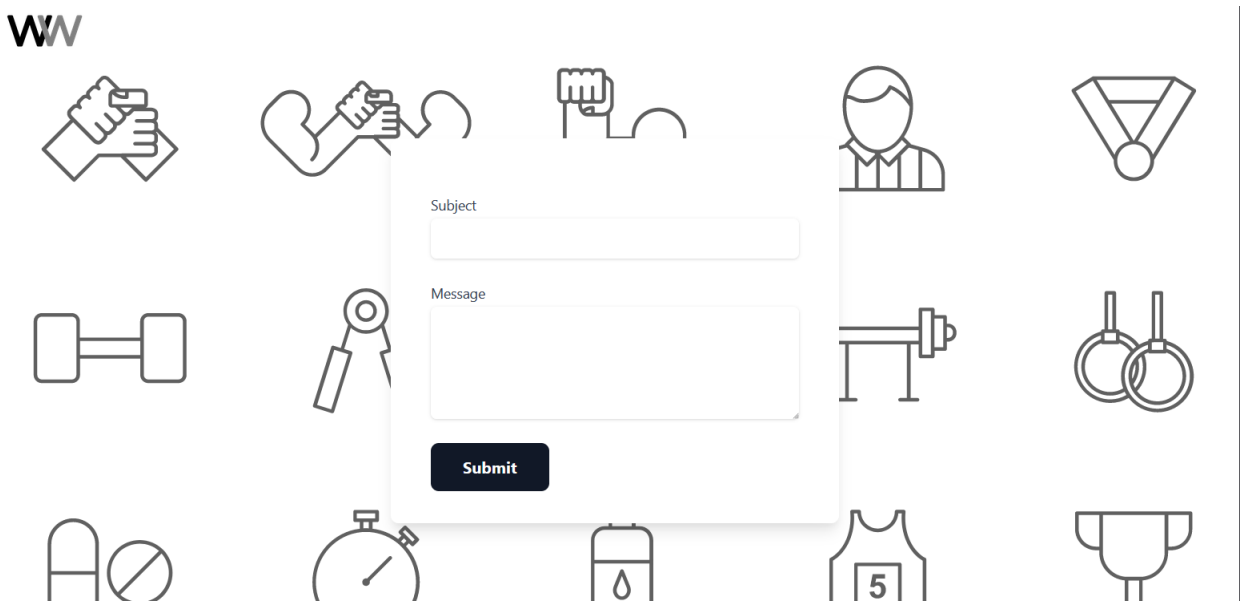


Figure 6.2: Contact Page

WodWide

is a social-utility platform design for cross training athletes and trainers

Meet the developers

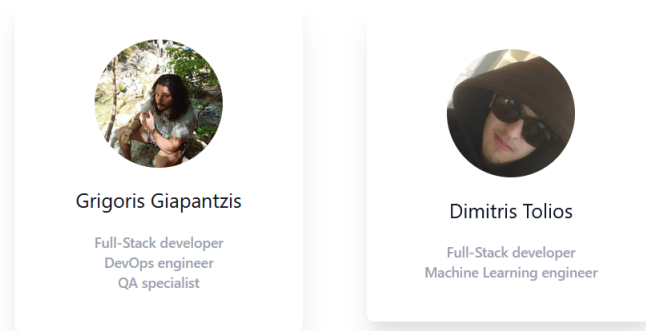


Figure 6.3: About Page

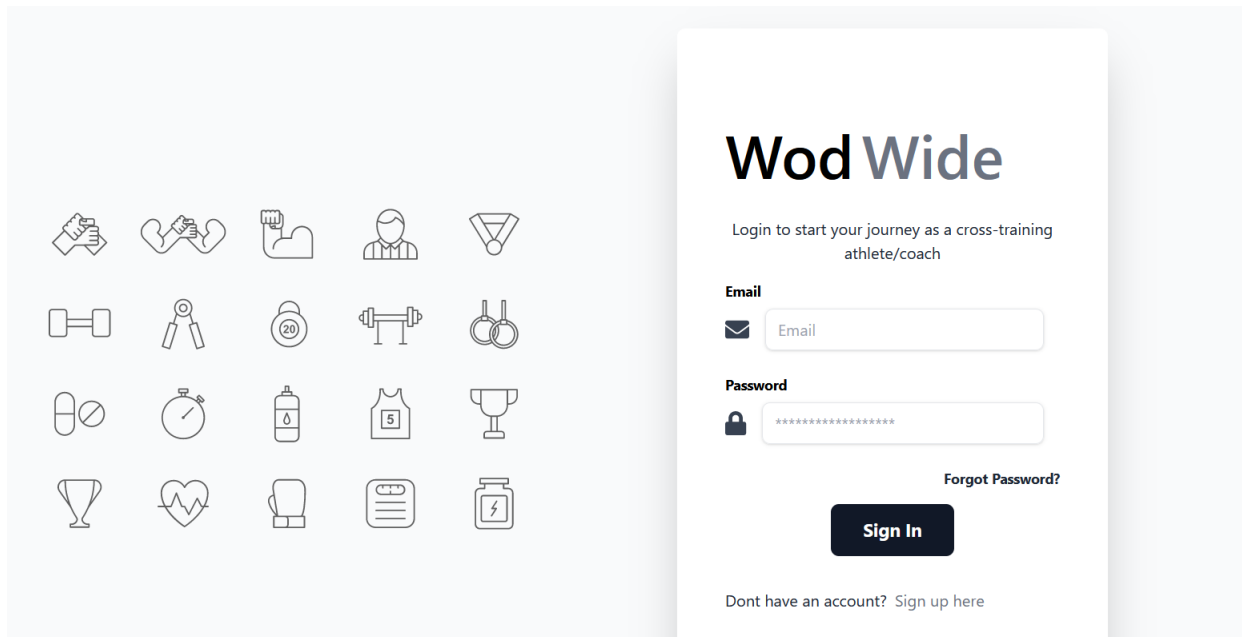
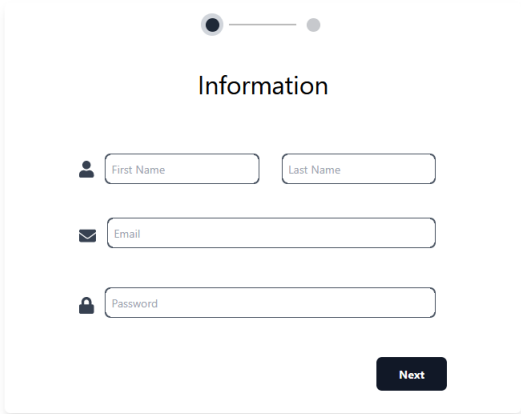


Figure 6.4: Login Page

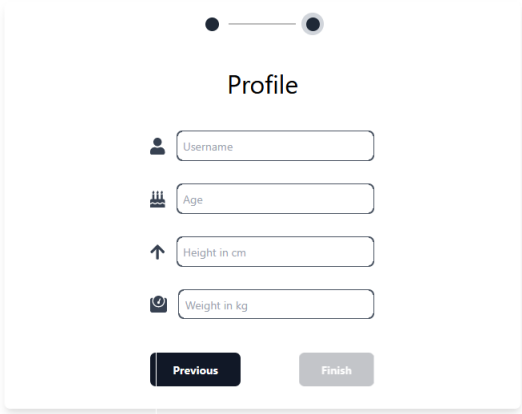
WW



The image shows a registration form titled "Information". At the top, there is a progress indicator with two dots, the first of which is filled. Below the title, there are four input fields: "First Name" and "Last Name" (side-by-side), "Email", and "Password". A "Next" button is located at the bottom right of the form.

Figure 6.5: Register Page: Step 1

WW



The image shows a registration form titled "Profile". At the top, there is a progress indicator with two dots, the second of which is filled. Below the title, there are four input fields: "Username", "Age", "Height in cm", and "Weight in kg". At the bottom, there are two buttons: "Previous" and "Finish".

Figure 6.6: Register Page: Step 2

The registration page consists of two steps. In the first step[Figure 6.5], users are required to enter their email, password, and full name. Once completed, they can proceed to the second step[Figure 6.6]. In the second step, users have the option to set their username, provide information such as their height, weight, and age.

After successfully logging in, users are redirected to the Dashboard page[Figure 6.7]. On the left side navigation bar of the dashboard, users will find several options to choose from. Additionally, on the map displayed on the dashboard, the pins represent the locations of registered gyms. By clicking on the Glossary page[Figure 6.8], users can access a collection of common CrossFit terms that are frequently used in the fitness community. This section provides definitions and explanations for these terms, helping users to better understand and familiarize themselves with the CrossFit terminology.

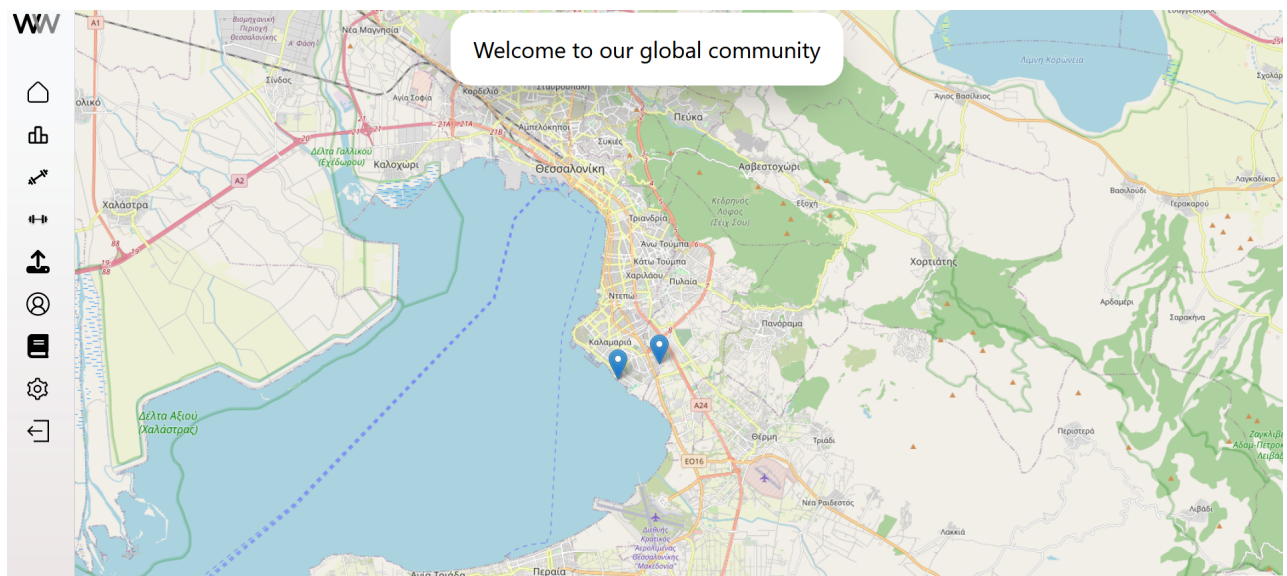


Figure 6.7: Dashboard Page

On the WOD (Workout of the Day) page[Figure 6.9], users have the option to select the equipment of their choice. Based on the selected equipment, the system will generate a customized workout specifically tailored to the available equipment. This allows users to create a workout that aligns with the equipment they have access to, providing them with a personalized and effective exercise routine.

In the account page[Figure 6.10], users have the ability to update their personal details and information. This includes modifying and editing information such as their name, username, height, weight and age. Users can easily make changes or updates to ensure their account information is accurate and up to date.

In the Settings page[Figure 6.11], users are provided with the option to delete their account if they wish to do so. This feature allows users to permanently remove their account and associated data from the system. It is an important decision and users should exercise caution before proceeding with the deletion, as it cannot be undone and all their account information will be permanently erased.

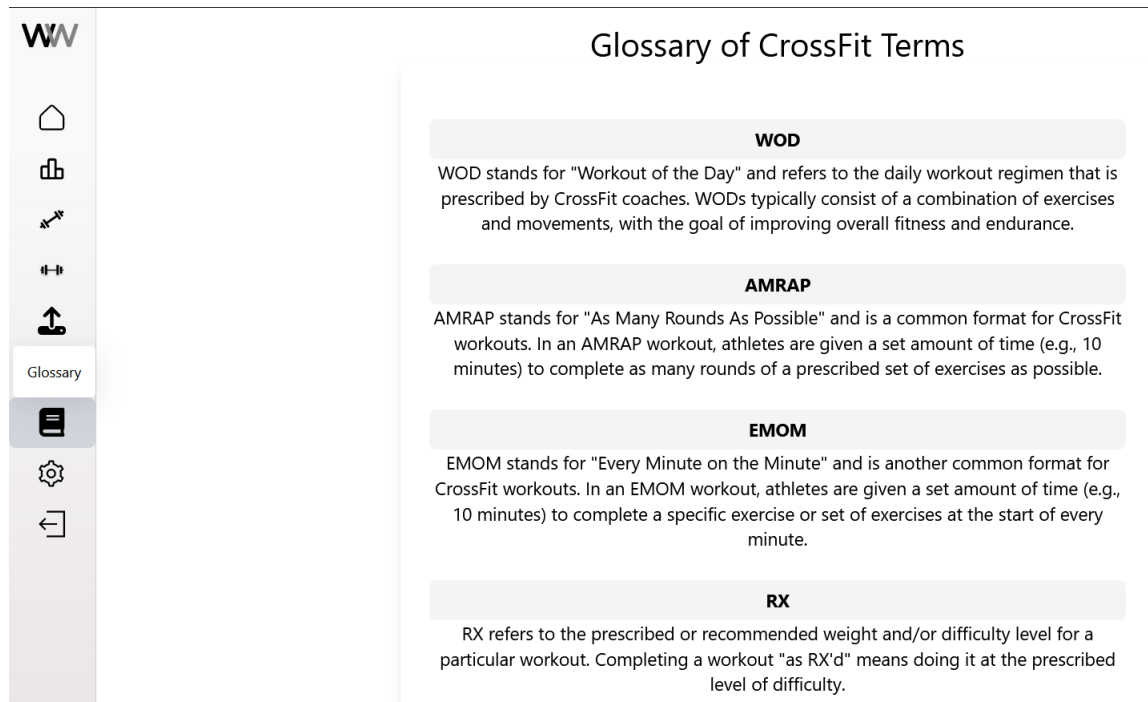


Figure 6.8: Glossary Page

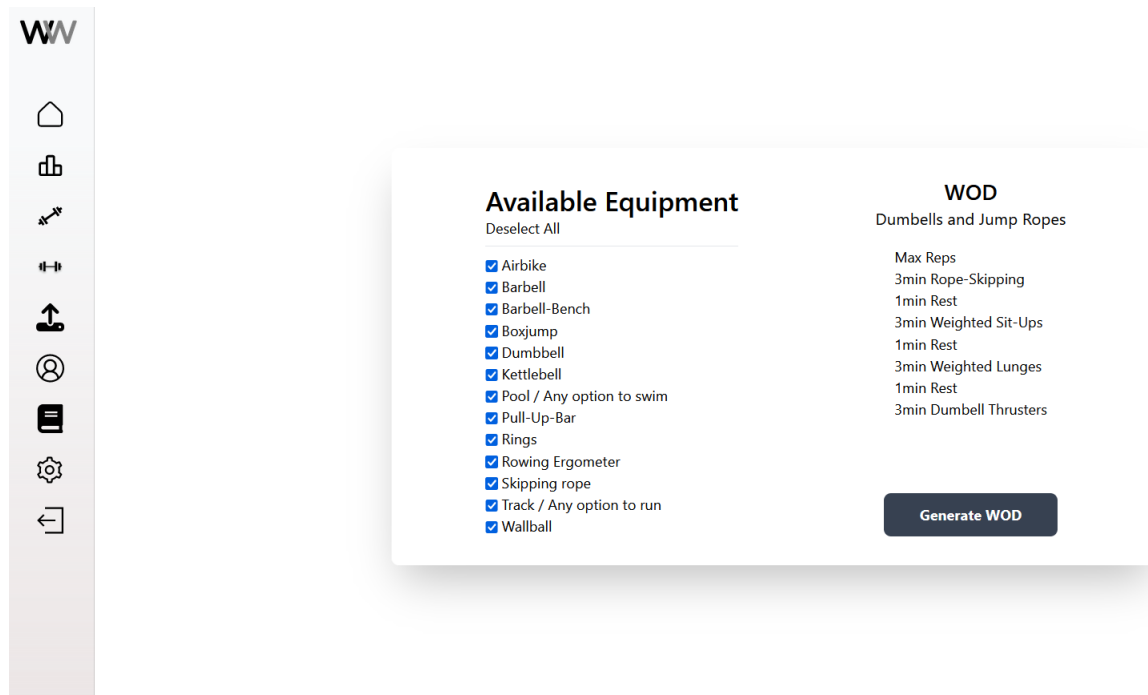


Figure 6.9: Wod Page

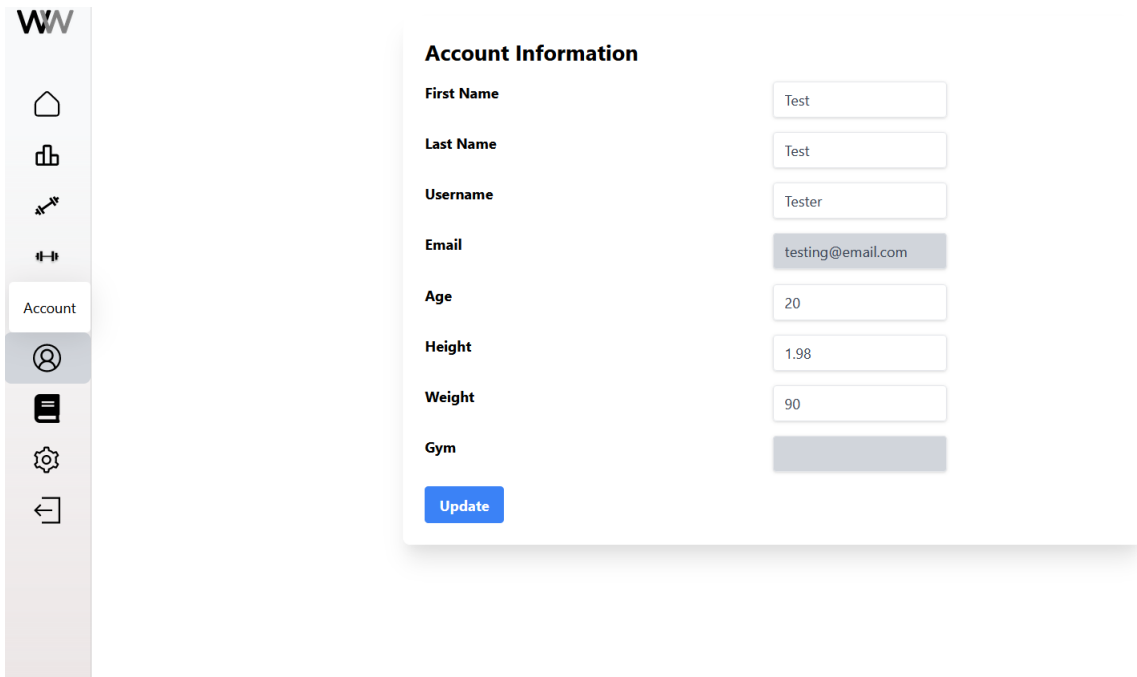


Figure 6.10: Account Page

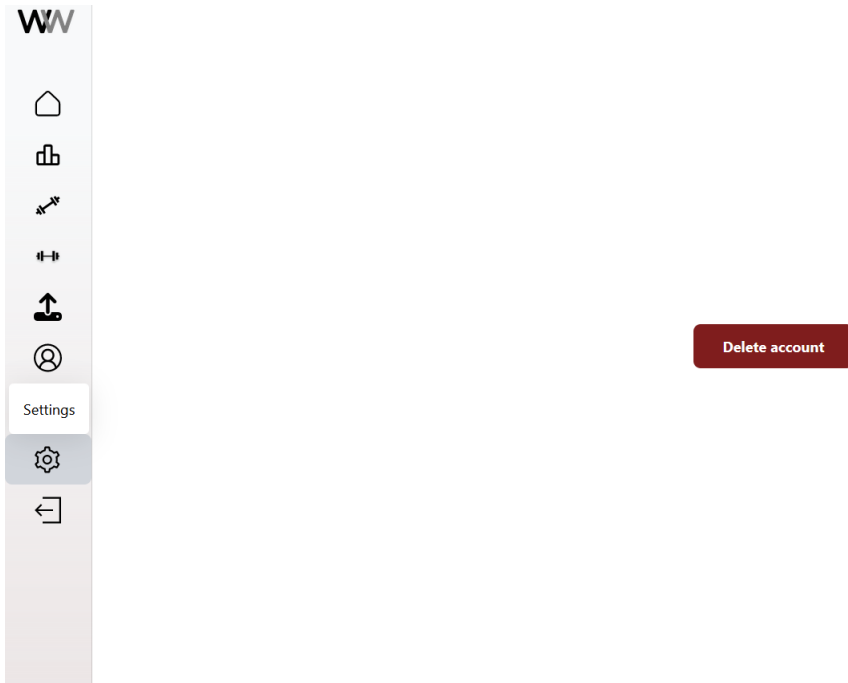


Figure 6.11: Account Page

When the users clicks on the logout button[Figure 6.12], they will be redirected back to the home page. This action effectively ends their current session and logs them out of the application, ensuring their account is securely signed out and no longer accessible until they log in again. After the user clicks on

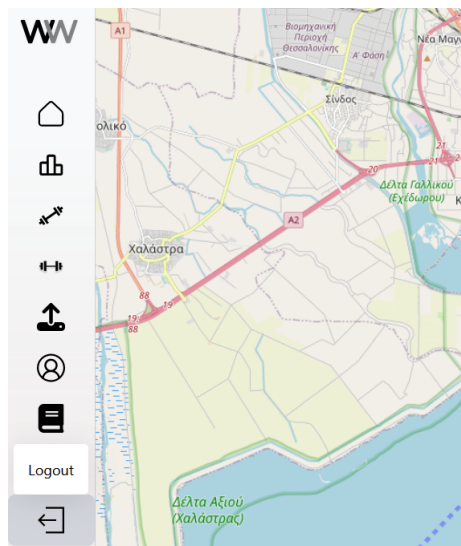


Figure 6.12: Logout button

”AI-submission,” they will be redirected to the video-submission page[Figure 6.13]. On this page, they can select the type of submission they want to make. If they choose ”count reps,” it will calculate the total number of reps for a single exercise. However, if they select ”for time,” they will have the option to define the number of reps for each exercise and can add more than one exercise to their workout. Additionally, they can specify the number of rounds they will perform with the same reps and exercises, allowing for a structured and customisable workout routine. Additionally, the video-submission page includes a time cap feature, which allows the user to set a maximum time limit for their submission. Once the defined time elapses, the submission process will automatically stop. Furthermore, users have the option to import and upload a video of their workout. When they submit the video, it gets uploaded to a Content Delivery Network (CDN) for efficient storage and retrieval. After the video is successfully uploaded, the frontend of the application receives a response containing the saved URL of the video. This URL can then be sent to the backend for further processing or storage if needed.

Once the video is assessed and the reps are counted, a modal will appear displaying the results[Figure 6.17]. This modal will show the total number of reps completed as well as the time taken to complete the exercise.

On the Gym page[Figure 6.18], users have the option to become a member as either a coach or an athlete in the platform. They can choose to join as a coach, providing their expertise and offering guidance to athletes, or they can join as an athlete, participating in workouts and training programs.

Additionally, users also have the option to create their own gym on the platform. This allows them to establish their own fitness facility, customize their training programs, and invite athletes to join their gym. It provides a platform for both gym owners and athletes to connect, collaborate, and benefit from the features and services offered by the platform.

Video submission

Welcome to the video submission page, where you can choose one or multiple exercises to submit your video for evaluation by an AI. We offer a variety of exercises for you to select from. Once you've made your selection, simply upload your video, and our AI system will assess it.

Type:

Exercise: Reps:

+ add exercise Total time cap:

Drag and drop your video
or
Click to add

Figure 6.13: Video submission Page

Video submission

Welcome to the video submission page, where you can choose one or multiple exercises to submit your video for evaluation by an AI. We offer a variety of exercises for you to select from. Once you've made your selection, simply upload your video, and our AI system will assess it.

Type: Rounds:

Exercise: Reps:

Exercise: Reps:

+ add exercise Total time cap:

pull-squats.mp4

Submit

Figure 6.14: Video submission Page: for time

Video submission

Welcome to the video submission page, where you can choose one or multiple exercises to submit your video for evaluation by an AI. We offer a variety of exercises for you to select from. Once you've made your selection, simply upload your video, and our AI system will assess it.

Type:

Exercise:

Total time cap:

Figure 6.15: Video submission Page: count reps

Video submission

Welcome to the video submission page, where you can choose one or multiple exercises to submit your video for evaluation by an AI. We offer a variety of exercises for you to select from. Once you've made your selection, simply upload your video, and our AI system will assess it.

Type:

Exercise:

Total time cap:

- pull-up
- push-up
- squat

Figure 6.16: Video submission Page: exercises

Video submission

Welcome to the video submission page, where you can choose one or multiple exercises to submit your video for evaluation by an AI. We offer a variety of exercises for you to select from. Once you've made your selection, simply upload your video, and our AI system will assess it.

Type:

Exercise:

Total time cap:

Results

Exercise: Time: Total reps:

Figure 6.17: Video submission Page: results

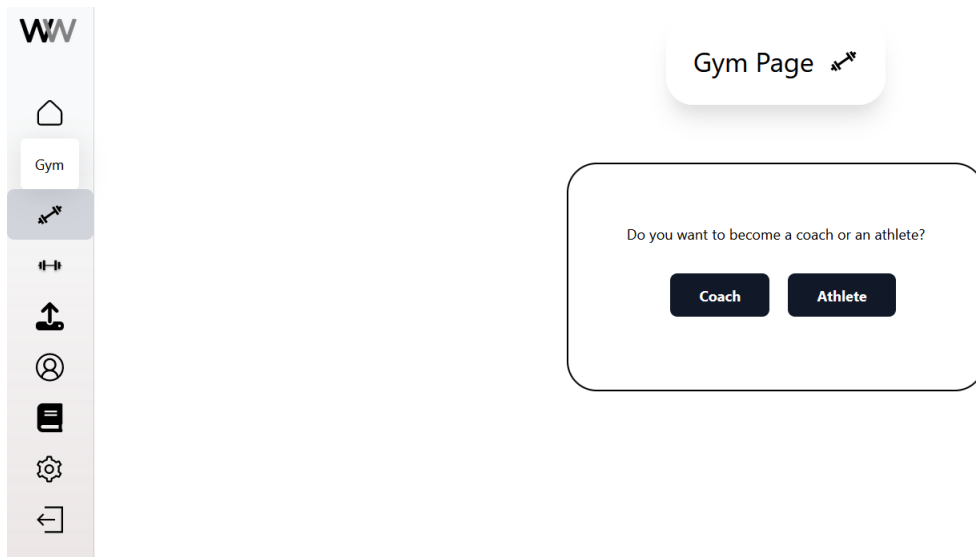


Figure 6.18: Gym Page

6.1 As a trainer

By clicking on "Coach" on the Gym page [Figure 6.18], users can begin the process of becoming a coach. The next step involves selecting between different tabs to determine their preference. They can choose to register their own gym and become an owner, establishing their own fitness facility and managing it independently [Figure 6.20]. Alternatively, they can opt to register within an existing gym [Figure 6.19], joining a pre-established fitness center and offering their coaching services within that gym's framework. The choice between these options allows users to tailor their coaching experience based on their goals and preferences. In the "Select Gym" tab, users have the ability to search for a gym using various criteria. They can search for a gym by its name, country, or phone number. This search functionality allows users to easily find and connect with a specific gym of their choice. Whether they have a specific gym in mind or want to explore gyms in a particular country, they can conveniently filter their search and discover suitable options [Figure 6.19]. When a user selects a gym card, it triggers a flip animation where the card flips to reveal the backside containing information about the gym. On the backside of the card, users can find details such as the gym's name, location, contact information, facilities, services offered, and any other relevant information about the gym. This feature provides users with a quick and convenient way to access and review important information about a particular gym before making a decision or taking further action. By clicking on the "Join as a Coach" button, users will be redirected to the dashboard. At this stage, they will need to complete the registration process to finalize their coach membership.

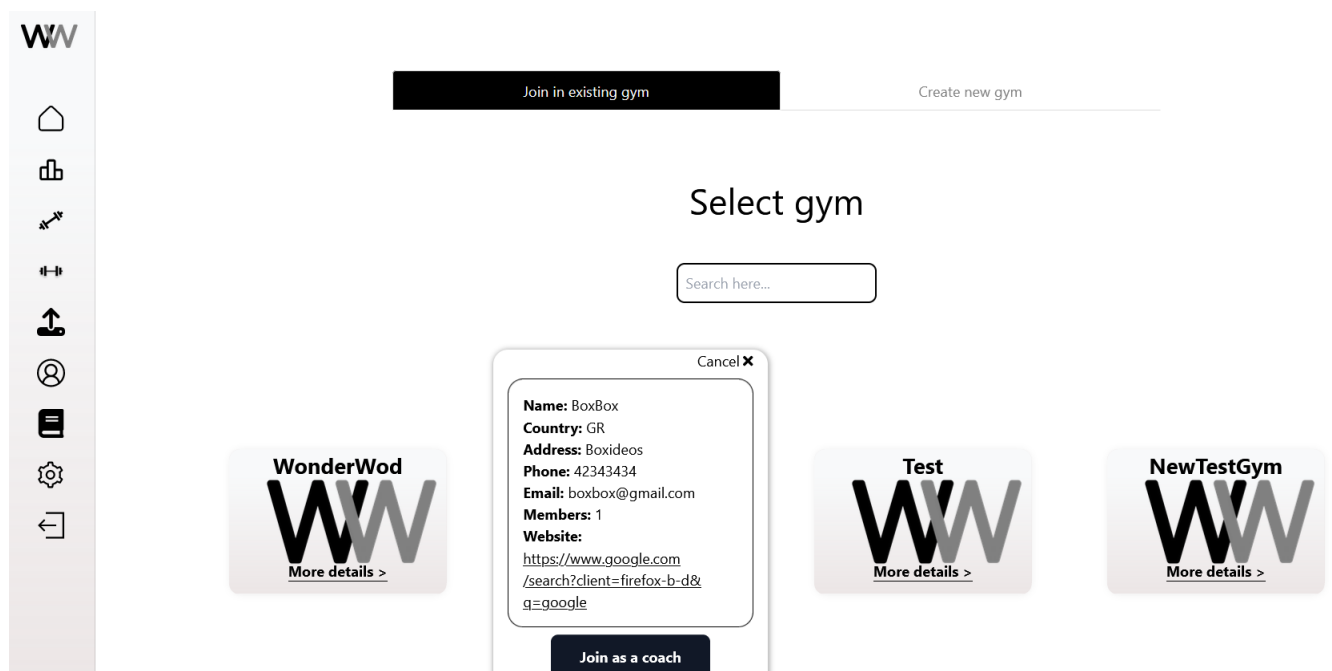


Figure 6.19: Select Gym

In the dashboard, there is a "Chat of the Day" feature where users from the same gym can communicate with each other [Figure 6.21]. This chat provides a dedicated space for gym members to interact, share information, ask questions, and engage in discussions related to their fitness journey. It fosters a sense of community and facilitates communication among gym members, allowing them to support and connect with one another within the platform.

WV

Join in existing gym Create new gym

Create gym

Submit

Figure 6.20: Create Gym

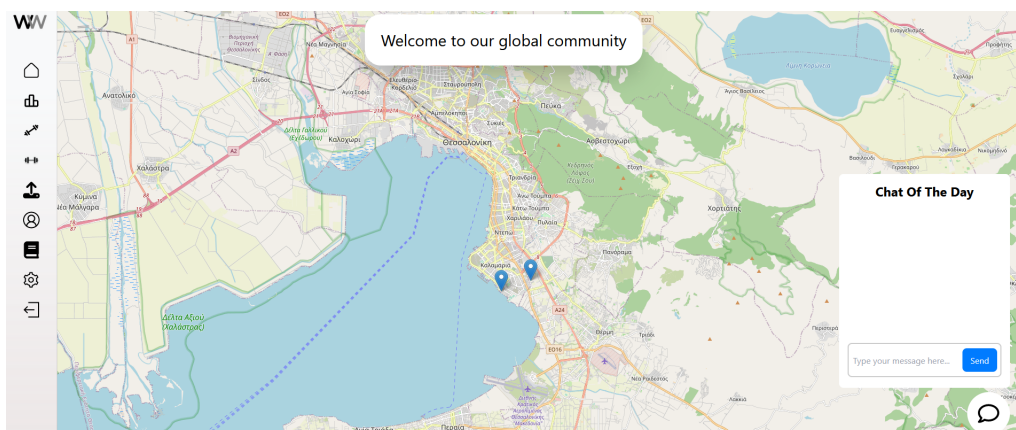


Figure 6.21: Chat of the Day

When accessing the gym page as a coach, users can view various details about the gym. On the left side of the page, they will find information such as the gym’s address, website, phone number, and country. On the right side, there will be staff-related information, including details about members, trainers, and the owner of the gym. In the middle of the page, coaches can access the WOD (Workout of the Day) section. In this section, coaches have the ability to upload new workouts or edit existing ones. This feature allows coaches to design and manage the daily workouts for the gym, ensuring that members have access to up-to-date and engaging training programs[Figure 6.22].

By selecting the "Update WOD" option, coaches can access an MD (Markdown) editor where they can write and update the Workout of the Day (WOD)[Figure 6.23]. The MD editor provides a user-friendly interface where coaches can format the workout description, add instructions, specify exercise details, and include any other relevant information using Markdown syntax. This allows coaches to easily create and edit the WOD content in a structured and visually appealing format[Figure 6.24].

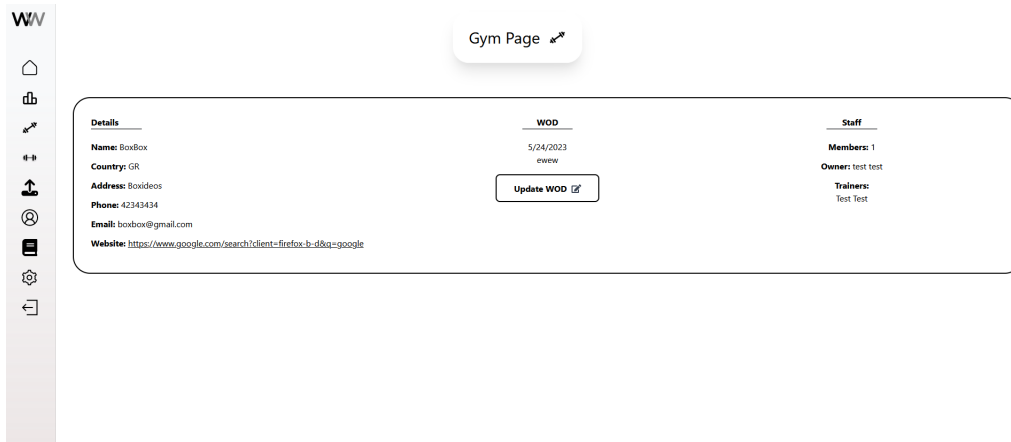


Figure 6.22: Gym Page: as a coach

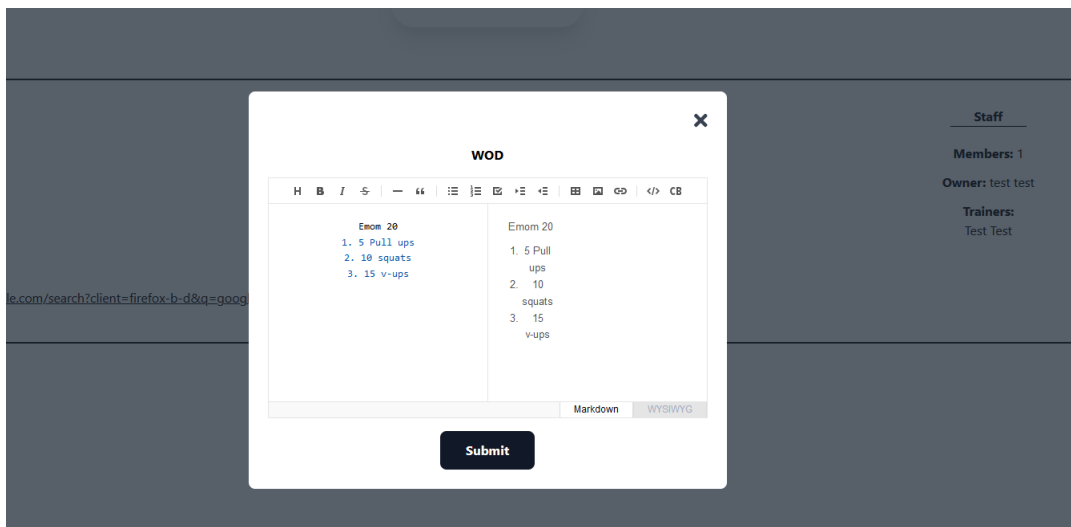


Figure 6.23: Gym Page: edit wod

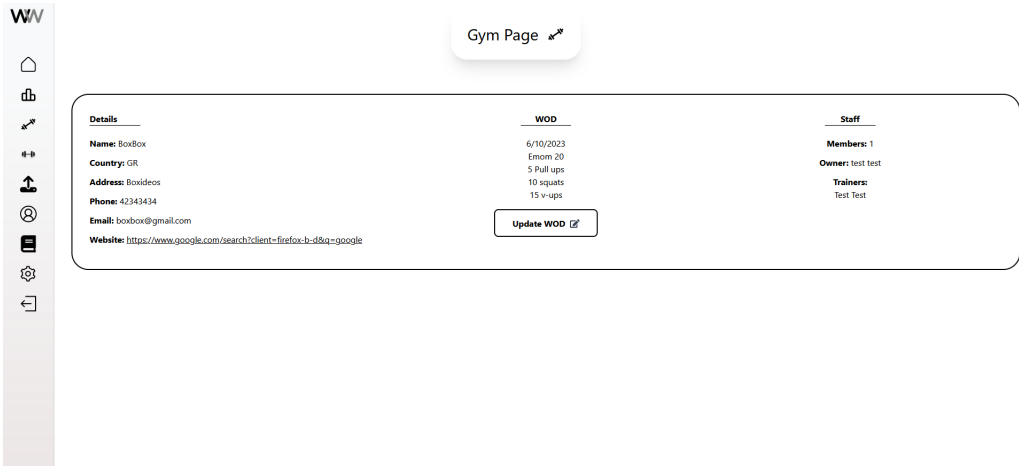


Figure 6.24: Gym Page: updated wod

6.2 As an athlete

By clicking on "Athlete" on the Gym page [Figure 6.18], users can initiate the process of becoming an athlete member. Similar to the coach process, they have the option to search for a gym and explore different gyms based on their preferences. When they find a gym of interest, they can click on the gym card to flip it and reveal detailed information about the gym on the backside.

After reviewing the information, if the user decides to join as a member, they can click on the "Join as a Member" button. By doing so, they will become a registered athlete member of the chosen gym, allowing them to access the gym's facilities, participate in workouts, and benefit from the services provided by the gym [Figure 6.25]. After becoming an athlete member, users will be redirected to the dashboard. In the dashboard, they will have access to the "Chat of the Day" feature. This feature allows athletes from the same gym to communicate and interact with each other. Athletes can engage in conversations, share tips, discuss workouts, ask questions, and provide support within this chat. It serves as a platform for gym members to connect, foster a sense of community, and build relationships with fellow athletes who train at the same gym [Figure 6.21].

In the leaderboard page [Figure 6.26], athletes can find the scores of the WOD (Workout of the Day) for registered gym members. They can view and compare the scores achieved by other athletes who are members of the same gym. This provides a competitive element and allows athletes to track their performance in relation to others. Additionally, athletes have the option to submit their own scores for the WOD. They can enter their results and submit them to be included in the leaderboard, allowing them to showcase their achievements and see how they stack up against other members of their gym.

By clicking the "Add Your Time" button, a modal will appear, presenting athletes with the option to input their time and provide an optional description [Figure 6.27]. Within this modal, athletes can enter the time it took them to complete the workout or WOD. They can also include an optional description, allowing them to add any additional notes or comments about their performance.

This feature enables athletes to contribute their results to the leaderboard and share relevant details regarding their workout experience. It promotes transparency, encourages healthy competition, and provides a platform for athletes to showcase their progress and performance within the gym community [Figure 6.28].

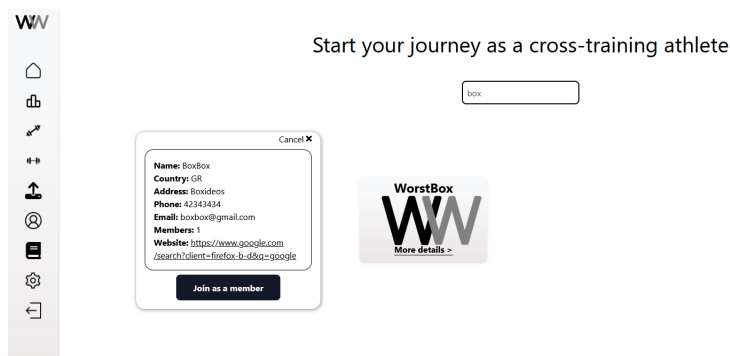


Figure 6.25: Gym Page: Join as athlete

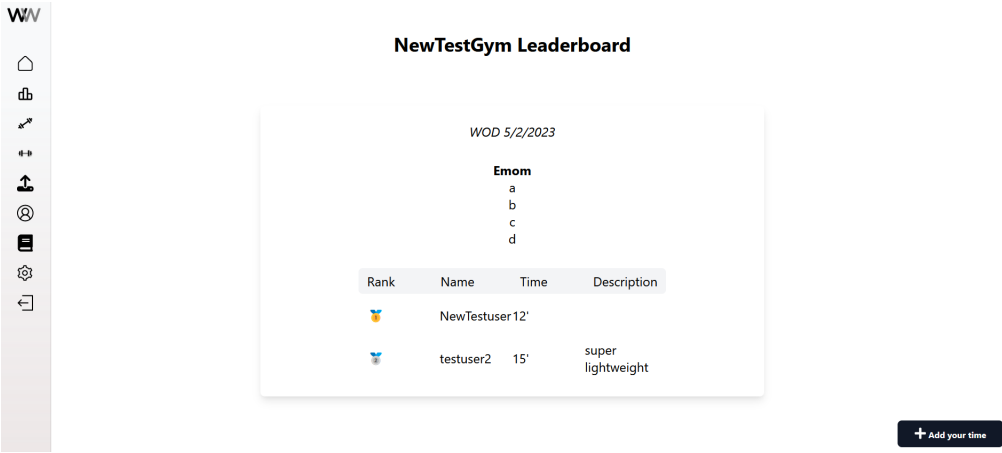


Figure 6.26: Leaderboard Page

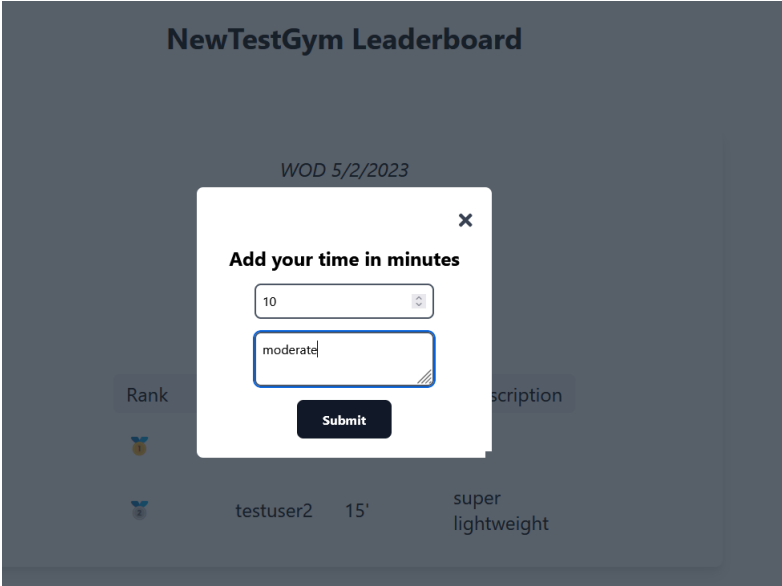


Figure 6.27: Leaderboard Page: submit score

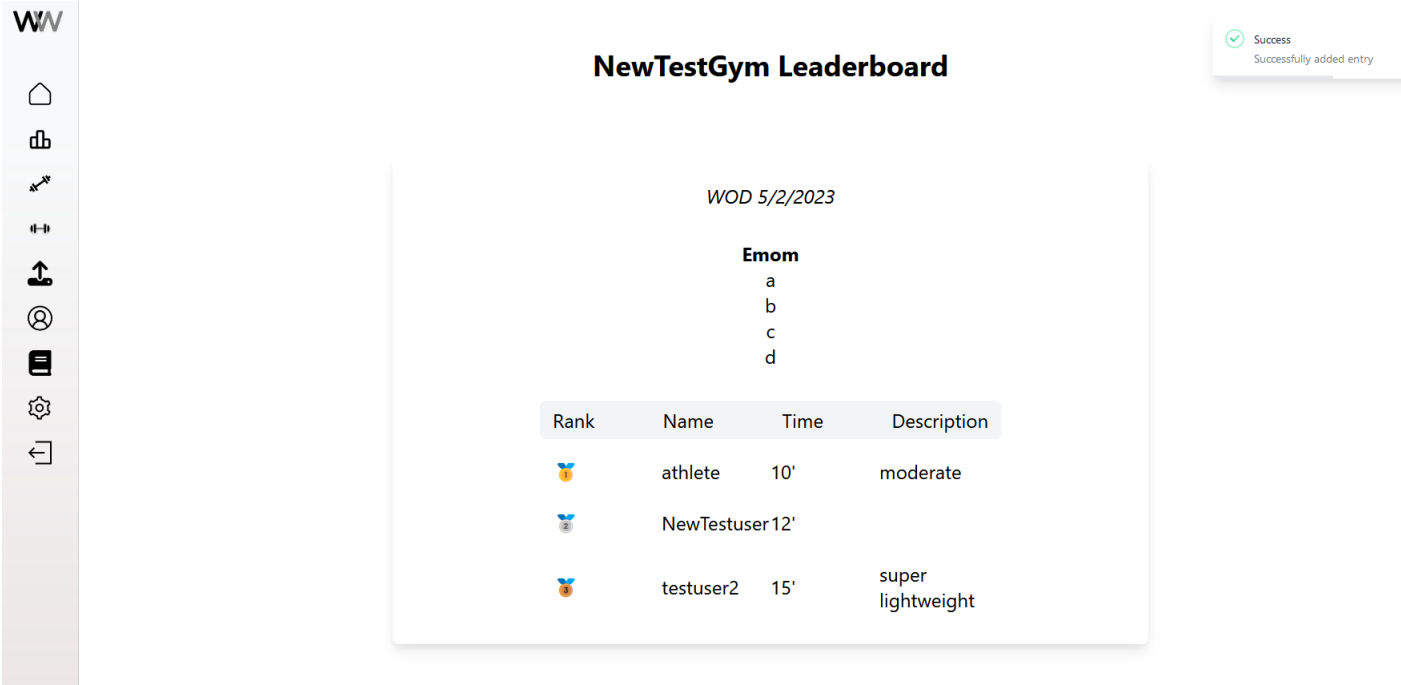


Figure 6.28: Leaderboard Page: updated

On the gym page, as an athlete, you will have the ability to view information about the gym. This means that you can access and read the details about the gym, such as its address, facilities, contact information, and any other relevant information. Similarly, you can also view the WOD for the day, which includes details about the workout, exercises, instructions, and any specific notes. However, as an athlete, you won't have the ability to edit or modify the WOD. You can simply read and follow the instructions provided to participate in the prescribed workout for the day[Figure 6.29].

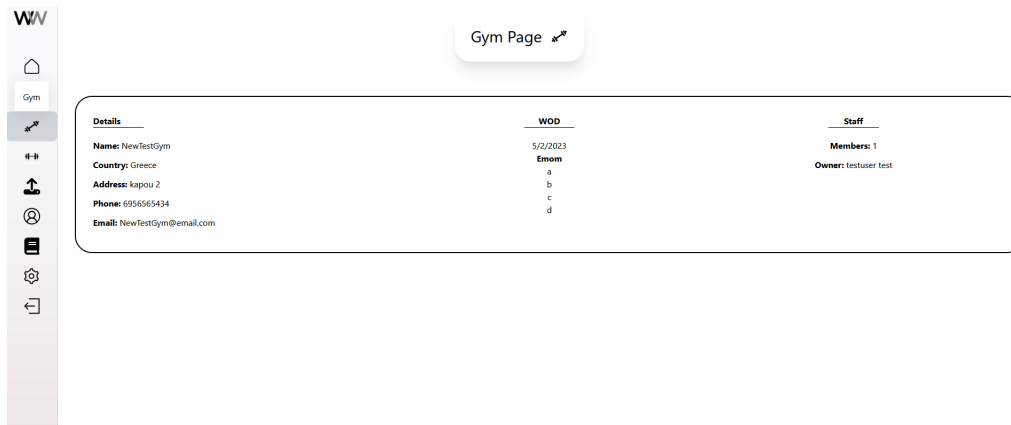


Figure 6.29: Gym Page: athlete

6.3 Video Evaluation

For video evaluation purposes, the FastAPI server handles requests from the frontend, which include the exercise details, the video uploaded to a CDN (Content Delivery Network), and the type of evaluation required. There are two different types of evaluation:

- **Count:** In this type, the server will analyze the video and count the number of reps performed for the specified exercise. It will identify each repetition and provide a count of how many reps were completed in the video.
- **For Time:** This type of evaluation focuses on a single exercise within the same request. The server will assess the video and determine the time it took to complete the specified exercise. It measures the duration from the starting point to the finishing point of the exercise.

The FastAPI server processes the received data, including the video and exercise details, and applies the appropriate evaluation algorithms based on the chosen type. It utilizes the information to accurately assess the performance of the exercise and provides the desired evaluation results to the frontend.

MediaPipe is used, specifically for three exercises: pull-ups, push-ups, and squats. MediaPipe provides a pretrained model that can detect and track landmarks of the human body in videos. These landmarks represent specific points on the body, such as joints or key body parts. Using the landmark data provided by MediaPipe, algorithmic functions are applied to determine the performance of each exercise. These functions analyze the movement patterns, positions, and transitions of the landmarks to assess the execution of pull-ups, push-ups, and squats. Based on these evaluations, feedback or scores can be generated to provide insights into the form and correctness of the exercises performed in the videos.

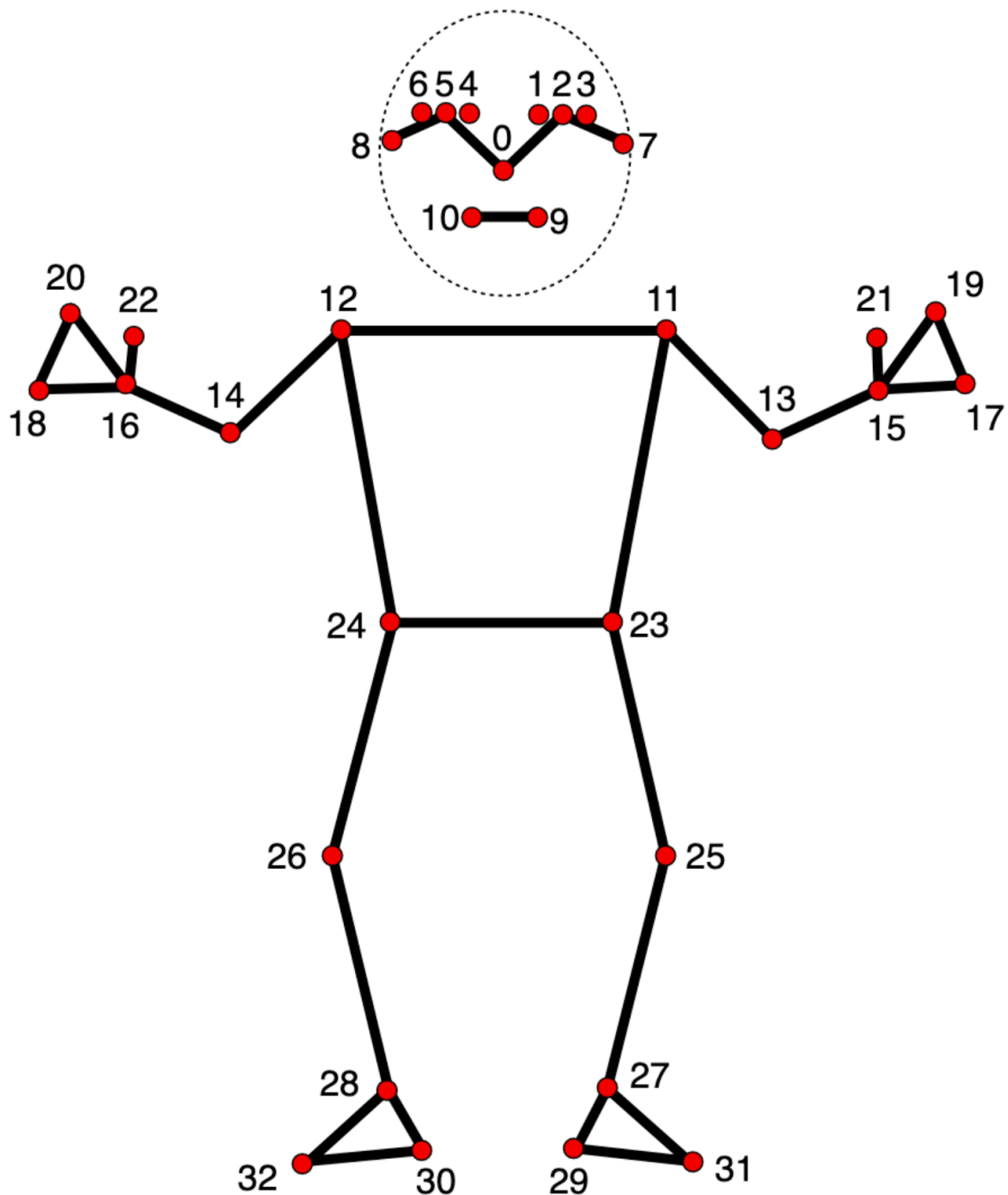


Figure 6.30: Media Pipe Landmarks

Chapter 7: Conclusion and future work

This thesis has presented the development of a groundbreaking web application, designed specifically for CrossFit athletes, that revolutionizes their training experience. By combining a comprehensive suite of features into a single platform tailored for the CrossFit community, this web app offers athletes the ability to discover and join local gyms, engage in meaningful conversations with fellow gym members, create customized workouts, and leverage artificial intelligence (AI) for accurate repetition verification.

Through extensive research, prototyping, and iterative testing, an intuitive and user-friendly interface was achieved, ensuring a seamless user experience. The integration of location-based services facilitates gym

discovery, empowering athletes to easily locate and explore nearby CrossFit facilities and connect with a supportive fitness community. The social interaction component fosters camaraderie and motivation by enabling athletes to connect, communicate, and share their experiences with like-minded individuals. Moreover, the workout creation feature empowers athletes to design personalized training regimens that align with their specific goals and preferences, providing a highly personalized and effective training experience.

One of the key innovations of this web app is the incorporation of AI technology for repetition verification. By leveraging computer vision algorithms, the app accurately analyzes and validates exercise repetitions performed by athletes, delivering real-time feedback on form and ensuring optimal technique. This unique capability eliminates the need for manual counting or subjective judgment, enhancing both the accuracy and objectivity of repetition tracking, and ultimately promoting safer and more efficient training practices.

Throughout the development process, valuable technical skills were honed in web development, AI integration, and user experience design. Additionally, a deeper understanding was gained of the unique requirements and challenges faced by CrossFit athletes, emphasizing the importance of creating a supportive and engaging online environment for their training journeys.

In conclusion, this thesis presents a pioneering web app for CrossFit athletes, offering a transformative platform that revolutionizes the way they discover gyms, interact with fellow athletes, create workouts, and verify repetitions. The combined functionalities of the app provide a holistic and empowering training experience, while the incorporation of AI technology enhances accuracy and safety. This project contributes to the advancement of technology in the fitness domain and is expected to greatly benefit the CrossFit community, inspiring future innovations in the field.

Future work:

- **Expanding the WOD List:** Continuously enriching the database with a wider range of workout options, including different fitness disciplines, specialized training programs, and popular workout trends. This would provide users with more diverse and personalized workout choices.
- **Extending ML Tool Capabilities:** Enhancing the ML tool's capabilities by expanding the list of available exercises for analysis. This can involve integrating additional exercise recognition models, leveraging deep learning techniques, and continuously updating the exercise database to ensure accurate and reliable results.
- **Mobile Application Development:** Considering the development of a companion mobile application for WodWide, allowing users to access the platform on-the-go, track workouts using mobile devices, and receive personalized notifications and reminders.

BIBLIOGRAPHY

- [1] K. Gurney, *An introduction to neural networks*. CRC press, 1997.
- [2] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.
- [3] R. Rojas and R. Rojas, “The backpropagation algorithm,” *Neural networks: a systematic introduction*, pp. 149–182, 1996.
- [4] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: Lstm cells and network architectures,” *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
- [5] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.
- [6] Google LLC, “Media Pipe.” https://developers.google.com/mediapipe/solutions/vision/pose_landmarker, January 2023. Accessed: May 1, 2023.
- [7] Vue.js team, “Vue Style Guide.” <https://v2.vuejs.org/v2/style-guide/?redirect=true>, December 2013. Accessed: April 15, 2023.
- [8] Pablohpsilva, “Vue.js Component Style Guide.” <https://github.com/pablohpsilva/vuejs-component-style-guide>, May 2022. Accessed: April 22, 2023.
- [9] Vuejs team, “ESLint Vue Plugin.” <https://eslint.vuejs.org/>, January 2019. Accessed: April 22, 2023.
- [10] Vuejs team, “Vue Patterns.” <https://learn-vuejs.github.io/vue-patterns/>, March 2021. Accessed: April 27, 2023.
- [11] Vladimir Starkov, Varya Stepanova, “BEM guides.” <https://getbem.com/>, August 2014. Accessed: April 22, 2023.
- [12] Sass team, “Sass: Syntactically Awesome Style Sheets.” <https://sass-lang.com/>, 2006. Accessed: April 22, 2023.
- [13] Google LLC, “Firebase.” <https://firebase.google.com/>, April 2012. Accessed: April 20, 2023.
- [14] FastAPI team, “FastAPI.” <https://fastapi.tiangolo.com/>, 2018. Accessed: April 22, 2023.
- [15] Nuxt.js team, “Nuxt test utils.” <https://nuxt.com/docs/getting-started/testing>, August 2020. Accessed: April 15, 2023.
- [16] Christian Bach, Mathias Biilmann, “Netlify.” <https://www.netlify.com/>, 2014. Accessed: April 22, 2023.
- [17] Cypress.io, “Cypress.” <https://www.cypress.io/>, February 2015. Accessed: June 2, 2023.