



NIB301CEM iOS APPLICATION DEVELOPMENT

Course Work

Full Name: Tharindu Sachin Wanniarachchi

NIBM Index No: COBSCCOMP211P-003

Coventry Index No: 11664651

Date of Submission: 21 - 05 - 2023

BSc (Hons) Computing 21.1 School of Computing, National Institute of Business Management School of Computing Colombo 07, Sri-Lanka.

Table of Contents

Table of (Contents	ii
Acknowle	edgement	1
Introducti	ion	1
Developn	nent	2
Front-End Development		2
1)	Welcome Page	2
2)	User Details Capture Page	3
3)	Home Page	4
4)	Notification Page	5
5)	Exercise Detail Page	6
6)	Suggest Fitness Plans	7
7)	View Custom Schedule	8
Back-End Development		9
1)	Web API Project	9
2)	Database	9
Other Development Tools		10
1)	Render	10
2)	Postman	10
3)	Source Control	11
Using Best Practices		12
1)	MVC Architecture	12
2)	Reusable Custom UI Component	13
3)	App Theme in Separate File	14
4)	Manageable Code Structure When Design Interfaces	15
5)	Created Separated Files to Manage API Services and Maintain Errors	16
Conclusio	on	18
Attachments		19
References		20

Acknowledgement

I am very thankful to our iOS lecturer Mr. Kosala Jayasekara who provided support and immense knowledge to complete this work successfully. Kosala Jayasekara sir's learning style and immeasurable coordination were very helpful in completing these tasks successfully. Once again, I thank you for your cooperation in successfully completing these tasks in a very short time and offer this to you with great respect.

Introduction

Exercise is essential for maintaining a healthy lifestyle. Also maintaining a proper BMI level and exercising the body properly can lead to a healthy lifestyle and a healthy body. By accessing this iOS mobile app, people can access workout plans based on the data they provide and can customize it if they want. Apart from that we can track our exercise progress. It is difficult to do with the busy lifestyle we live nowadays but the ios app makes it very easy to do. This iOS mobile app provides thorough knowledge on choosing the right exercise poses and the proper way to perform them.

Development

Front-End Development

For this I used Swift 5.7 and UIKit with programmatic UI to create this iOS mobile app. I developed it using macOS version 12.6.1 and Xcode version 14.1 as my IDE.

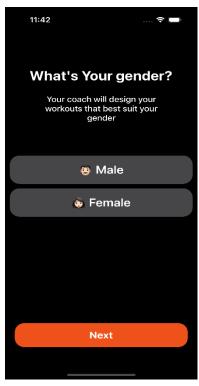
1) Welcome Page

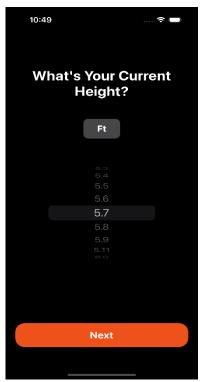
When the user accesses this iOS mobile application, the first thing user sees this welcome screen. This screen acts as the initial interface and introduces the application.

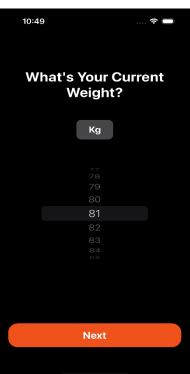


2) User Details Capture Page

These Onboard Pages are used to capture the user's Age, Gender, Height, Weight, Fitness Goal, and details. Based on the data provided by the user, this fitness application calculates the BMI index and provides the fitness plans user needs.









3) Home Page

This is the homepage of this mobile application, which shows the exercise details.

- Shows the details of the user logged into the application.
- Shows user's exercise history details.
- Shows a list of categorized exercises.

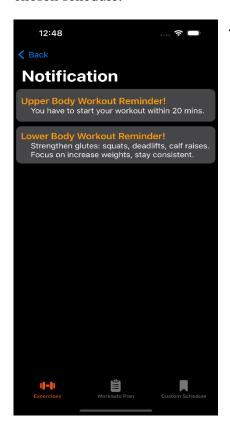






4) Notification Page

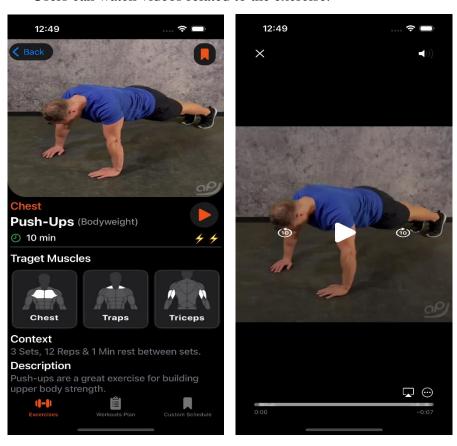
The app can send reminders and notifications to complete their workouts based on the user's chosen schedule.



5) Exercise Detail Page

Further details of an exercise are displayed on the Exercise Detail page

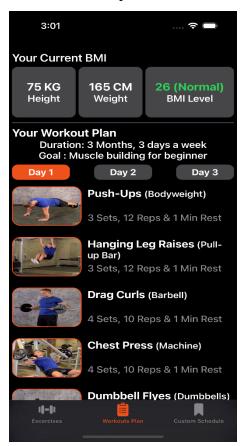
- Shows the exercise name, category, set & rep count, time, and target muscles other details.
- Users can add exercise to custom schedule if desired.
- Users can watch videos related to the exercise.



6) Suggest Fitness Plans

This page calculates the BMI index based on the data entered by the user and suggests a fitness plan.

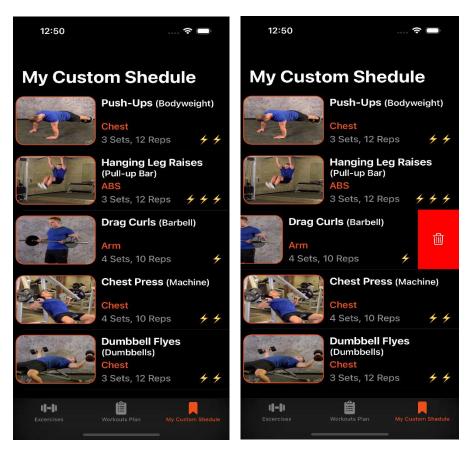
- Displays the user's height weight and BMI value.
- The workout plan that suits the user is displayed. It is three days for a week.



7) View Custom Schedule

This page displays the exercises that the user added from the exercise detail page.

- The exercises in the user's custom schedule are displayed.
- User can remove an exercise by swiping a table cell to the left in the exercise list.



Back-End Development

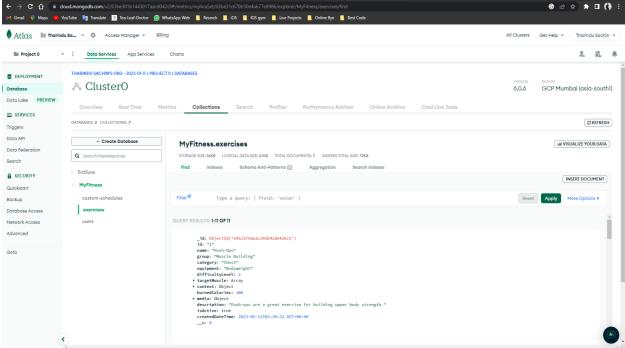
1) Web API Project

Node.js and Express framework was used to create the backend REST API project. Created models and Routes for user, exercise and customSchedule.

```
Ð
                                      user.js ...\routes X user.js ...\routes
                                           routes > 🖪 customSchedule.js > 😚 customScheduleRouter.get("/:userId") callback
                                             const express = require("express");
                                             const async = require("async");
                                             const customScheduleRouter = express.Router();
const customScheduleModel = require("../models/customSchedule");
         middlewares
                                             const exerciseModel = require("../models/exercise");
                                          8 > customScheduleRouter.post("/", async (req, res) => { ...
           customSchedule.js
exercise.js
          user.js
                                         74 customScheduleRouter.get("/:userId", async (req, res) => {
                                                  const customScheduleExercises = await customScheduleModel
                                                     userId: req.params.userId,
                                                    .sort({
    createdDateTime: "desc",
                                                   if (customScheduleExercises.length === 0) {
                                                         success: false,
                                                         message:
                                                           "The given user does not match any custom schedule exercises on our system",
                                                         data:
```

2) Database

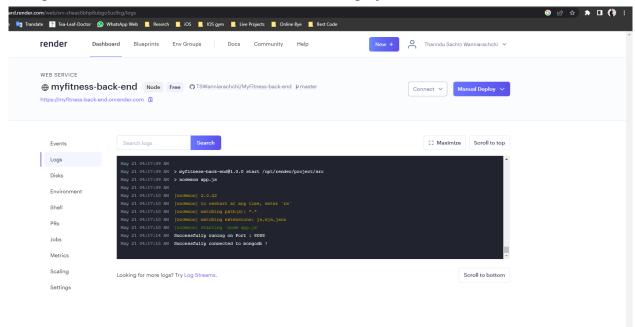
The MongoDB Atlas cloud database was used as the backend database for the project.



Other Development Tools

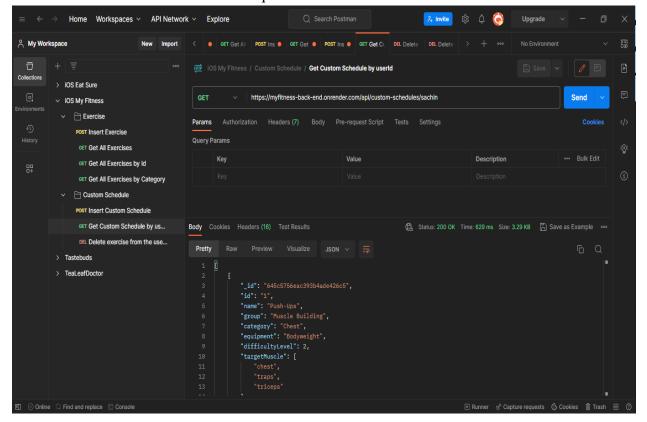
1) Render

This platform was used to host the RESTful web API project.



2) Postman

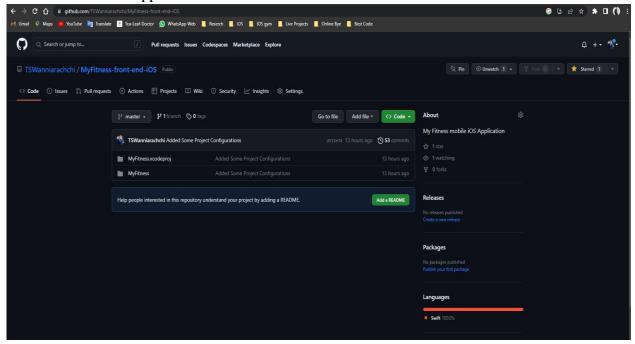
Postman was used to check the API requests



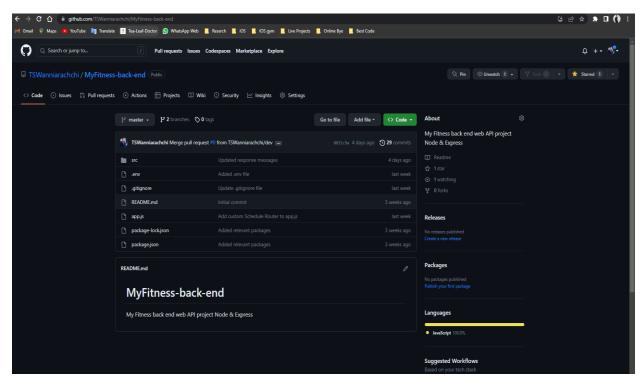
3) Source Control

Used GitHub as a source control tool for web API project and iOS mobile application.

i iOS mobile application



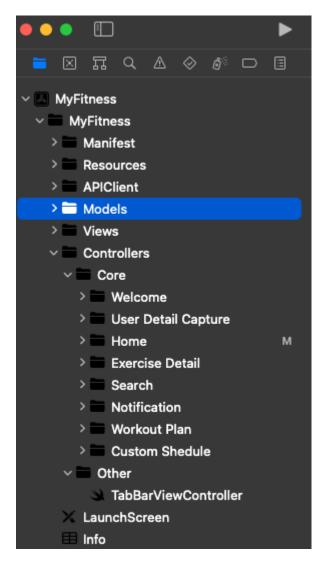
ii Node.js web API project



Using Best Practices

1) MVC Architecture

MVC architecture was used to create this application. It helps organize code properly by separating the view (user interface), models (database), and containing logic in controllers.



2) Reusable Custom UI Component

A reusable custom component was created during the creation of this application. It made it easy to create UI.

Below is the reusable UI Component and how it is used in a view controller.

```
₩ MyFitness
MyFitness > [] iPhone 14 Pro
                                                                                               Indexing | Processing files
                                                                                                                                      ₹ 10 1 00
                                                      HomeExerci...wController
                                                                               CustomButton
                                                                                                    WorkoutPla...wController
                                         MyFitness > 	≡ MyFitness > ≡ Views > ≡ Components > 	≤ CustomButton > □ CustomButton
    > Manifest
                                            10 class CustomButton: UIButton {
    > Resources
                                                   enum FontSize {
    > APIClient
    > Models
                                                       case big
                                                       case med
                                                       case small
     ∨ Components
         CustomLabel
                                                   init(buttonType: ButtonType, title: String, fontsize: FontSize){
    super.init(frame: .zero)
          CustomButton
          CustomHorizontalBar
          CustomSpinner
          CustomlmageButton
                                                       self.translatesAutoresizingMaskIntoConstraints = false
    Controllers
     ∨ ■ Core
                                                       switch buttonType{
       > Welcome
                                                       case .primary:
    self.backgroundColor = ColorGuide.primary
       > User Detail Capture
                                                           self.setTitleColor(.label, for: .normal)
       ∨ ■ Home
                                                       case .secondary:
            HomeExer...Controller M
                                                          self.backgroundColor = .tertiaryLabel
            ExerciseLis...tionViewCell
            HomeWork...tionViewCell
                                                          self.layer.borderColor = ColorGuide.primary.cgColor
       > Exercise Detail
       > Search
        > Notification
```

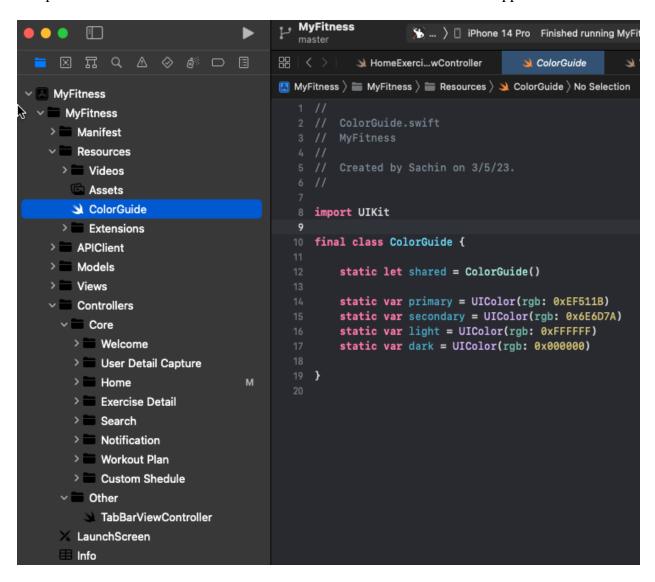
```
😘 ... 🖒 📋 iPhone 14 Pro Finished running MyFitness on iPhone 14 Pro

→ HomeExerciseViewController

                                                                                                                ₹ ≣0 | Œ
                                    CustomButton
                                                     🔼 MyFitness 🕽 🚞 MyFitness 🤇 🚞 Controllers 🤉 🚞 Core 🤇 🚞 Home 🕽 🔌 HomeExerciseViewController 🕽 👩 HomeViewController
    import UIKit
    class HomeViewController: UIViewController {
        //MARK: - Variables
        var ExerciseData = [ExerciseModel]()
        private let userProfileImageView = CustomImageView(image: UIImage(systemName: "questionmark")!,
                                                      imageType: .avatar,
imageLayout: .light)
       textAlianment: .left)
        private let usernameLabel = CustomLabel(labelType: .header3,
                                            textColor: .label,
        private let stepLabel = CustomLabel(labelType: .card,
                                        textColor: .label,
textAlignment: .center)
        private let calorieBurnLabel = CustomLabel(labelType: .card,
                                              textAlignment: .center)
        private let progressLabel = CustomLabel(labelType: .card,
```

3) App Theme in Separate File

A separate file was maintained to store the theme color of this iOS mobile application.



4) Manageable Code Structure When Design Interfaces

When designing the user interface, created a separate function as shown below and call the viewDidLoad() function. It made the code very easy to maintain. As a best code practice, the tasks performed by the relevant function were commented on.

```
MyFitness
                                                            😘 ... 🖒 🗓 iPhone 14 Pro 🛘 Finished running MyFitness on iPhone 14 Pro
                                                                                                                                                                  CustomSheduleViewController
                                                                                     WorkoutPlanViewController
                                                                                                                                                           ≠ ≣□ | Œ

→ HomeExerciseViewController

                                                                                                                         SceneDelegate
🔼 MyFitness 〉 🚞 MyFitness 〉 🚞 Controllers 〉 🚞 Core 〉 🚞 Custom Shedule 〉 メ CustomSheduleViewController 〉 🔃 coverlmageView
   8 import UIKit
     class CustomScheduleViewController: UIViewController {
           //MARK: - Variables
           var CustomScheduleExerciseData = [ExerciseModel]()
           var ResponseData = [ResponseModel]()
           //MARK: - UI Components
private let spinner = CustomSpinner(size: .med,
                                                       tintColor: ColorGuide.primary)
  20
21
22
23
24
25
26
27
28
           private let coverImageView = CustomImageView(image: UIImage(systemName: "questionmark")!,
                                                                  imageType: .page,
imageLayout: .light)
                let tableView = UITableView()
                tableView.backgroundColor = .systemBackground
tableView.allowsSelection = true
                \textbf{table View.} \textbf{register} \textbf{(Custom Schedule Exercise Table View Cell.} \textbf{self,} \textbf{ for Cell Reuse I dentifier:}
                     CustomScheduleExerciseTableViewCell.identifier)
                tableView.translatesAutoresizingMaskIntoConstraints = false
                return tableView
                                                            😘 ... 🕽 🗓 iPhone 14 Pro 🛮 Finished running MyFitness on iPhone 14 Pro 🕵
                                              CustomSheduleViewController

→ HomeExerciseViewController

                                                                                      SceneDelegate
🔼 MyFitness \rangle 🚞 MyFitness \rangle 🚞 Controllers \rangle 🚞 Core \rangle 🚞 Custom Shedule \rangle 🔌 CustomSheduleViewController \rangle 🔃 coverImageView
  10 class CustomScheduleViewController: UIViewController {
           //MARK: - Life Cycle
           override func viewDidLoad() {
    super.viewDidLoad()
                view.backgroundColor = .systemBackground
title = "My Custom Shedule"
                addSubviews()
                setUpValues()
           // MARK: - Add Subviews
           private func addSubviews(){
                view.addSubview(coverImageView)
                view.addSubview(customScheduleTableView)
           // MARK: - UI Setup Constraints
                     spinner.widthAnchor.constraint(equalToConstant: 100), spinner.heightAnchor.constraint(equalToConstant: 100),
                     spinner.centerYAnchor.constraint(equalTo: view.centerYAnchor),
                     coverImageView.topAnchor.constraint(equalTo: view.safeAreaLayoutGuide.topAnchor, constant: 0),
                    coverImageView.leftAnchor.constraint(equalTo: view.leftAnchor),
coverImageView.rightAnchor.constraint(equalTo: view.rightAnchor),
```

```
MyFitness
36 ... ) [] iPhone 14 Pro Finished running MyFitness on iPhone 14 Pro
                                                                                                                                 ≠ ≣0 | Œ

→ HomeExerciseViewController

                                         CustomSheduleViewController
                                                                        SceneDelegate
🄼 MyFitness 🕽 🚞 MyFitness 🤇 🚞 Controllers 🤇 🚞 Core 🤇 🚞 Custom Shedule 🕽 🔌 CustomSheduleViewController 🕽 💽 coverImageView
  10 class CustomScheduleViewController: UIViewController {
         // MARK: - Setup TableView
             customScheduleTableView.delegate = self
             customScheduleTableView.dataSource = self
         // MARK: - Setup Values
              coverImageView.image = UIImage(named: "cover_custom-schedule")
             visibleComponents(isVisible: false)
             fetchCustomScheduleExercisesData(userId: "sachin")
         private func fetchCustomScheduleExercisesData(userId: String) {
              // Create API request
             let request = Request(endpoint: .customSchedules,
                                   pathComponents: [userId])
             // Call API request & get response
APICaller.shared.getCustomSchedules(URL: request.url) { result in
                 DispatchQueue.main.asvnc {
                      switch result {
                      case .success(let model):
                          self.CustomScheduleExerciseData = model
```

5) Created Separated Files to Manage API Services and Maintain Errors

i Store End-Points

```
MyFitness
master
36 ... ) [] iPhone 14 Pro Finished running MyFitness on iPhone 14 Pro
                                                        HomeExerci...wController
                                                                                      EndPoint
                                                                                                      APICaller
                                                                                                                      🔌 WorkoutPla...wCon ⇄ 🗐 🛙 🖽
                                           MyFitness > \( \bigcirc \) MyFitness > \( \bigcirc \) APIClient > \( \bigcirc \) EndPoint > \( \bigcirc \) Endpoint
∨ MyFitness

∨ ■ MyFitness

   > Manifest
   > Resources

✓ APIClient

       EndPoint
                                   М
                                                 import Foundation
       Request
       @frozen enum Endpoint: String {

✓ ■ Models

       Exercise
                                                      case exercises
        Response
                                                     case customSchedules = "custom-schedules"
   > Views
   Controllers
     ∨ ■ Core
       v Welcome
            WelcomeViewController
       > User Detail Capture
```

ii Maintain API Request in separate file

```
MyFitness
                                                                ... \ [] iPhone 14 Pro Finished running MyFitness on iPhone 14 Pro
                                                   HomeExerci...wController
                                                                                  EndPoint
                                                                                                APICaller
                                                                                                                 MyFitness > ■ MyFitness > ■ APIClient > → APICaller > M getExercises(URL:completion:)
MyFitness
                                            10 final class APICaller {

∨ ■ MyFitness

   > Manifest
                                                    // MARK: - Exercises
   > Resources
                                                   public func getExercises(URL url: String, completion: @escaping
    (Result<[ExerciseModel], Error>) -> Void) {
   ∨ ■ APIClient
        EndPoint
        Request
       APICaller
   ∨ ■ Models
                                                           let task = URLSession.shared.dataTask(with: request) {
        Exercise
                                                                data, response, error in
        Response
                                                                guard let data = data, error == nil else{
   > Views
                                                                    completion(.failure(error ?? APIError.failedToGetData))
   ∨ Controllers
     ∨ ■ Core
            WelcomeViewController
                                                                    let result = try JSONDecoder().decode([ExerciseModel].self, from:
           User Detail Capture
                                                                        data)
                                                                    completion(.success(result))
            HomeExer...Controller M
           ExerciseLis...tionViewCell
                                                                    completion(.failure(error))
            HomeWork...tionViewCell
       > Exercise Detail
                                                            task.resume()
        > Search
       > Notification
        > Workout Plan
```

iii Fetching data from API call and error handlng

```
36 ... ) [] iPhone 14 Pro Finished running MyFitness on iPhone 14 Pro
                                                                                                                              M HomeExerci...iewController
                                      EndPoint
                                                   CustomShed...iewController
                                                                               WorkoutPla...ewController
                                                                                                          🙎 MyFitness 🖒 🚞 MyFitness 🕽 🚞 Controllers 🕽 🚞 Core 🤉 🚞 Custom Shedule 🕽 🔌 CustomSheduleViewController 🕽 🔝 fetchCustomScheduleExercisesData(userld:)
 10 class CustomScheduleViewController: UIViewController {
        private func fetchCustomScheduleExercisesData(userId: String) {
            APICaller.shared.getCustomSchedules(URL: request.url) { result in
                DispatchQueue.main.async {
                    switch result {
                    case .success(let model):
                        self.CustomScheduleExerciseData = model
                    case .failure(_):
                        let alert = UIAlertController(title: "Error",
                                                    message: String(describing: "Error occurred while fetching custom schedule
                                                        exercises."),
                        alert.addAction(UIAlertAction(title: "Dismiss",
                                                     style: .default,
handler: nil))
                        self.present(alert, animated: true, completion: nil)
```

Conclusion

This web application is a valuable tool to help people maintain their exercise and BMI level properly. By accessing this mobile app, people can access workout plans based on the data they provide, and they can also customize it if they want. It can also store and analyze the user's exercise data. This mobile application is designed using new technology and user-friendly interfaces are created using correct color combinations. This iOS mobile application helps the user to maintain physical and mental fitness and lead a healthy life.

Attachments

1) Back-end web API project GitHub link:

https://github.com/TSWanniarachchi/MyFitness-back-end

2) Front-end iOS mobile application GitHub link:

https://github.com/TSWanniarachchi/MyFitness-front-end-iOS

3) Application workflow video link:

https://drive.google.com/drive/folders/1urx0odu7e6ducZ-yW3C0OsjB8orlZbse?usp=share_link

4) All resources link:

https://drive.google.com/drive/folders/1buDY_mfwqfF_IXdQXAm3hJ9bGI-RfX7P?usp=share_link

References

Brain4Code Lerning (2021) *How to Fetch Data from WebAPI and display into Custom CollectionView in Swift 5 XCode*. [Online] Available from: https://www.youtube.com/watch?v=a-DgwFCV104/> [4th May 2023].

Chiladze, M. (2023) *Fitness App Design UI Shot*.[Online] Available from: https://dribbble.com/shots/20085391-Fitness-App-Design-UI-Shot/ [10th April 2023]

Danuarta, D. (2023) *Workout Planner App*.[Online] Available from: https://dribbble.com/shots/20399574-Workout-Planner-App/ [10th April 2023]

Giovani, F. (2023) *DEROFIT - Workout Planner*. [Online] Available from: https://dribbble.com/shots/20399747-DEROFIT-Workout-Planner/> [11th April 2023]

iOS Academy (2021) Swift: TableView w/ Custom Cells Tutorial (2021, iOS) - 2021. [Online] Available from: https://www.youtube.com/watch?v=R2Ng8Vj2yhY/> [14th May 2023].

iOS Academy (2020) Swift 5: Custom CollectionView Cells (Programmatically) Xcode 11 - 2020 iOS. [Online] Available from:

https://www.youtube.com/watch?v=mwsVA2gJTTM&t=3s/ [14th May 2023].