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

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

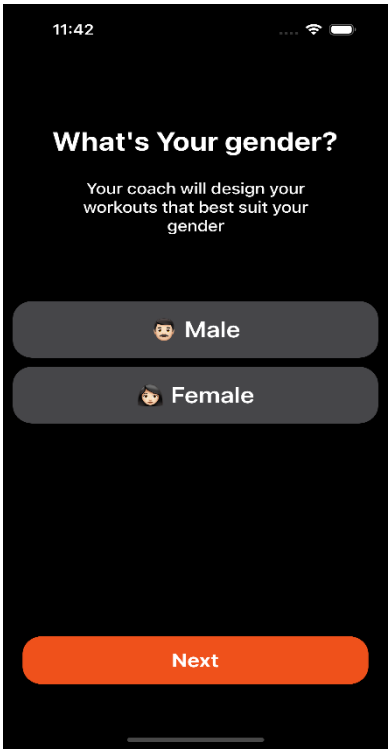
For this I used Swift 5.7 and UIKit with programmatic UI to create this iOS mobile app. I developed it using macOS version 12.6.1 and Xcode version 14.1 as my IDE.

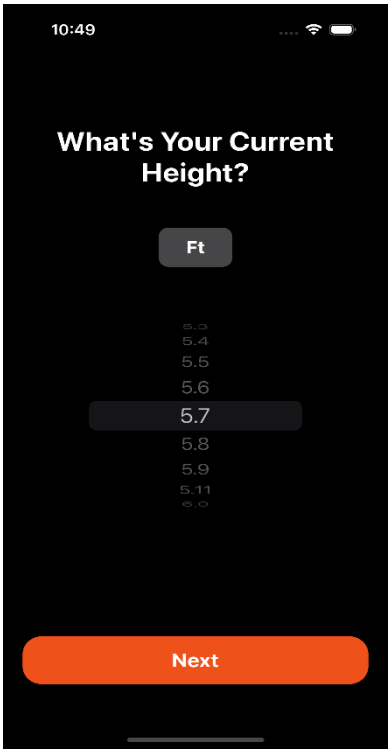
1) Welcome Page

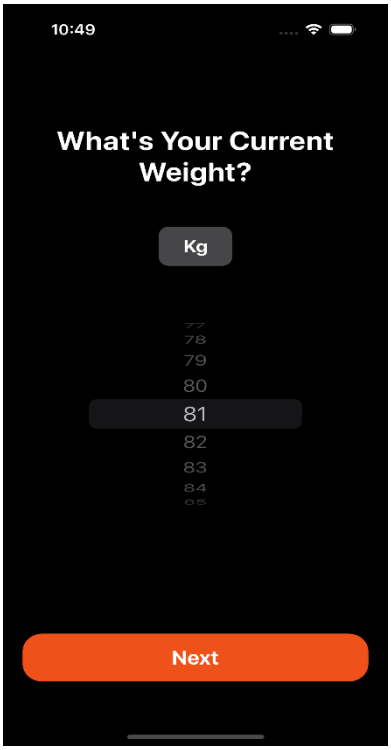
When the user accesses this iOS mobile application, the first thing user sees this welcome screen. This screen acts as the initial interface and introduces the application.

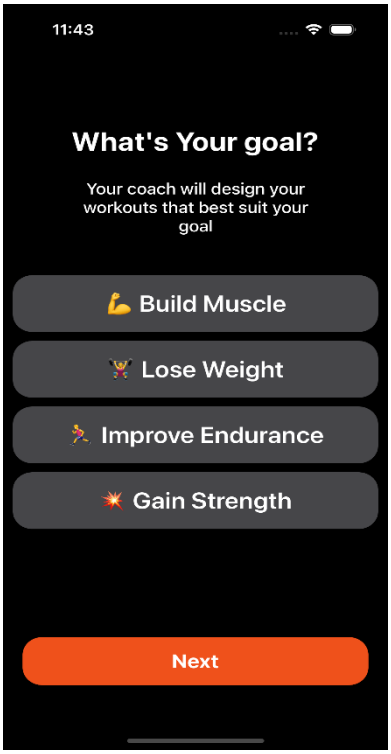
2) User Details Capture Page

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.





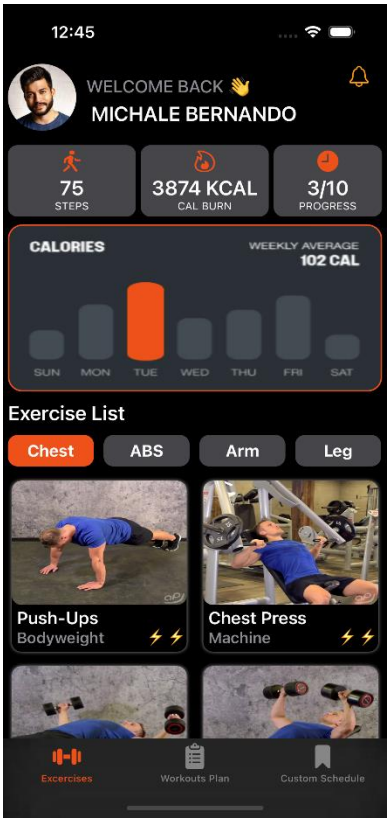


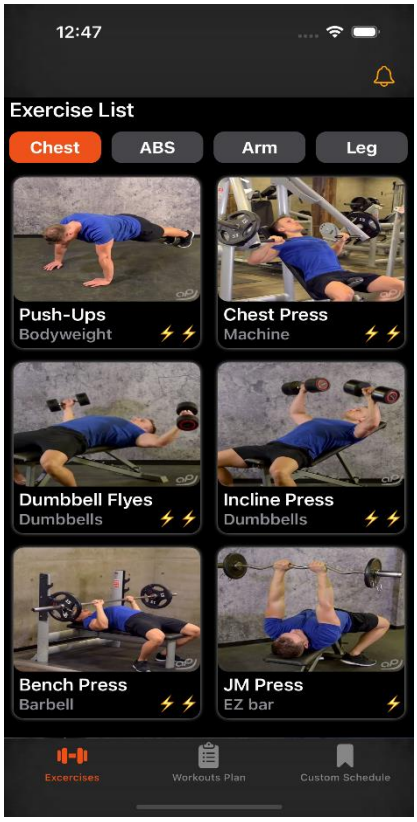


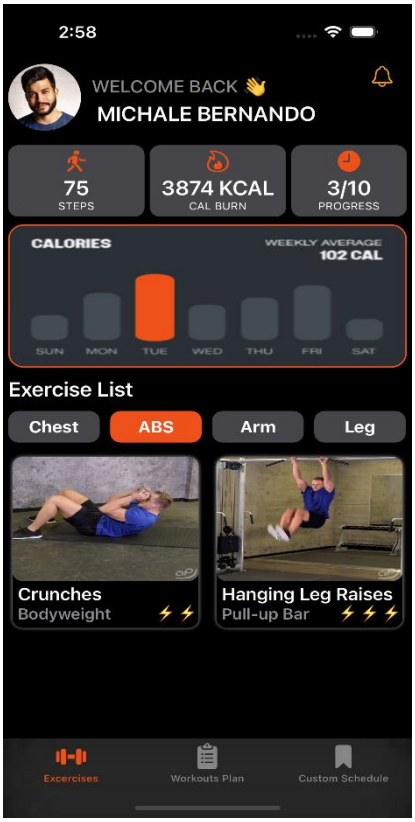
3) Home Page

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.

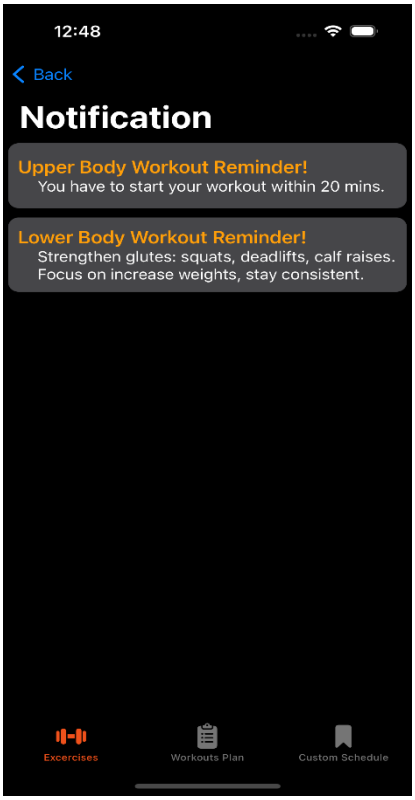






4) Notification Page

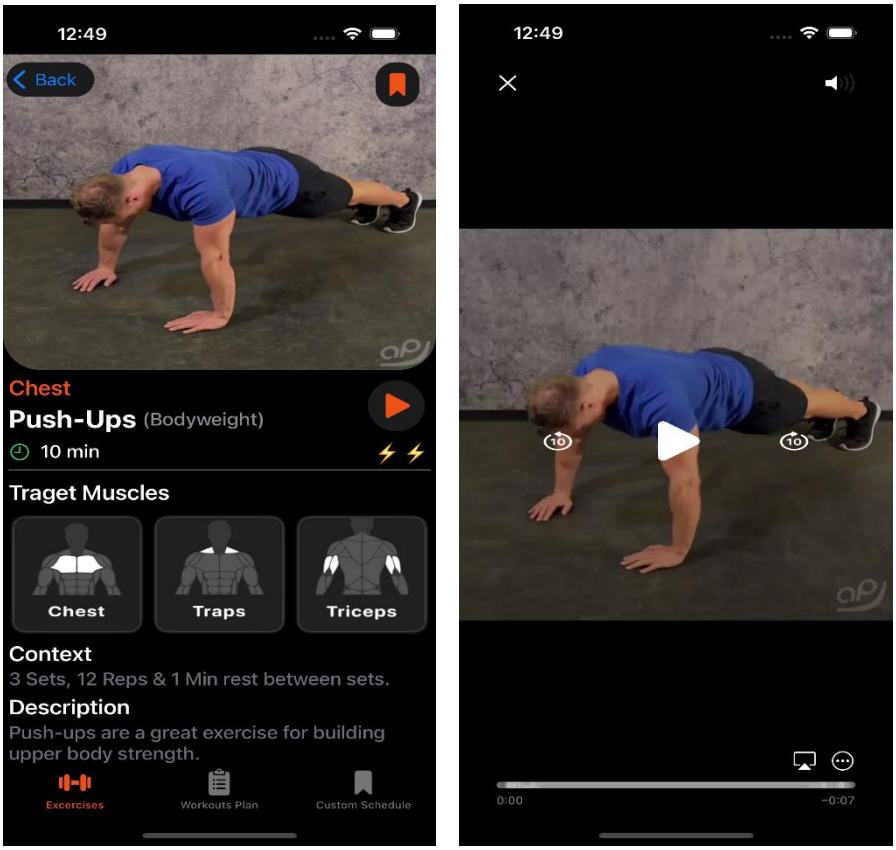
The app can send reminders and notifications to complete their workouts based on the user's chosen schedule.



5) Exercise Detail Page

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

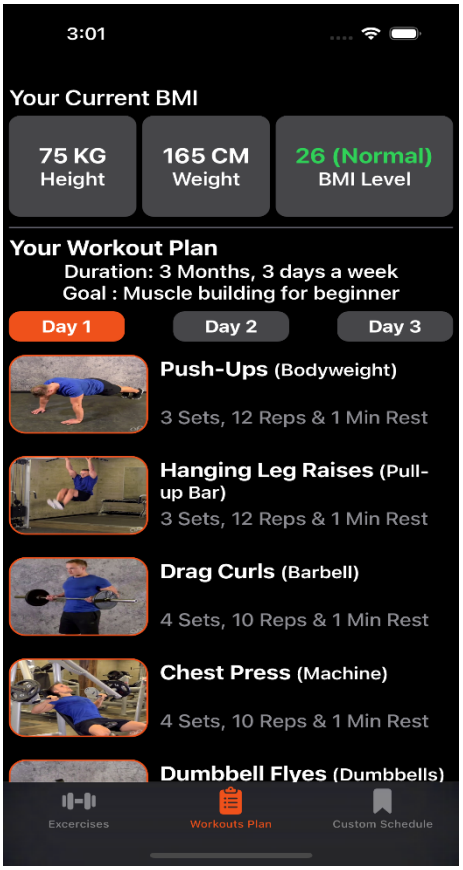
- Shows the exercise name, category, set & rep count, time, and target muscles other details.
- Users can add exercise to custom schedule if desired.
- Users can watch videos related to the exercise.



6) Suggest Fitness Plans

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

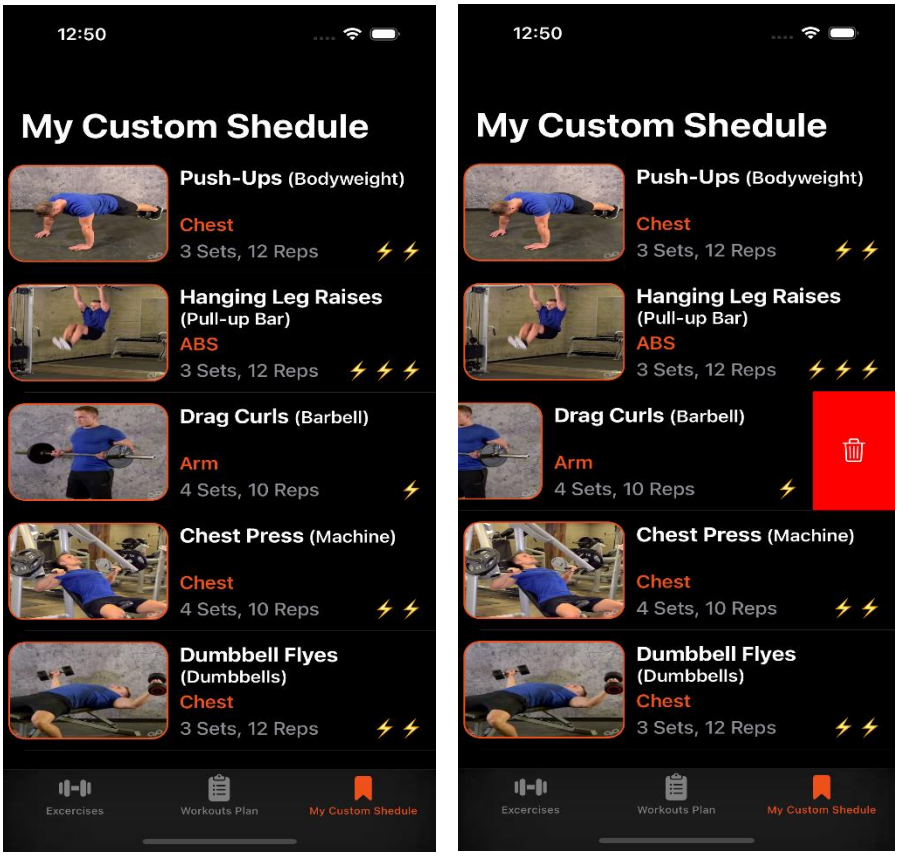
- Displays the user's height weight and BMI value.
- The workout plan that suits the user is displayed. It is three days for a week.



7) View Custom Schedule

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

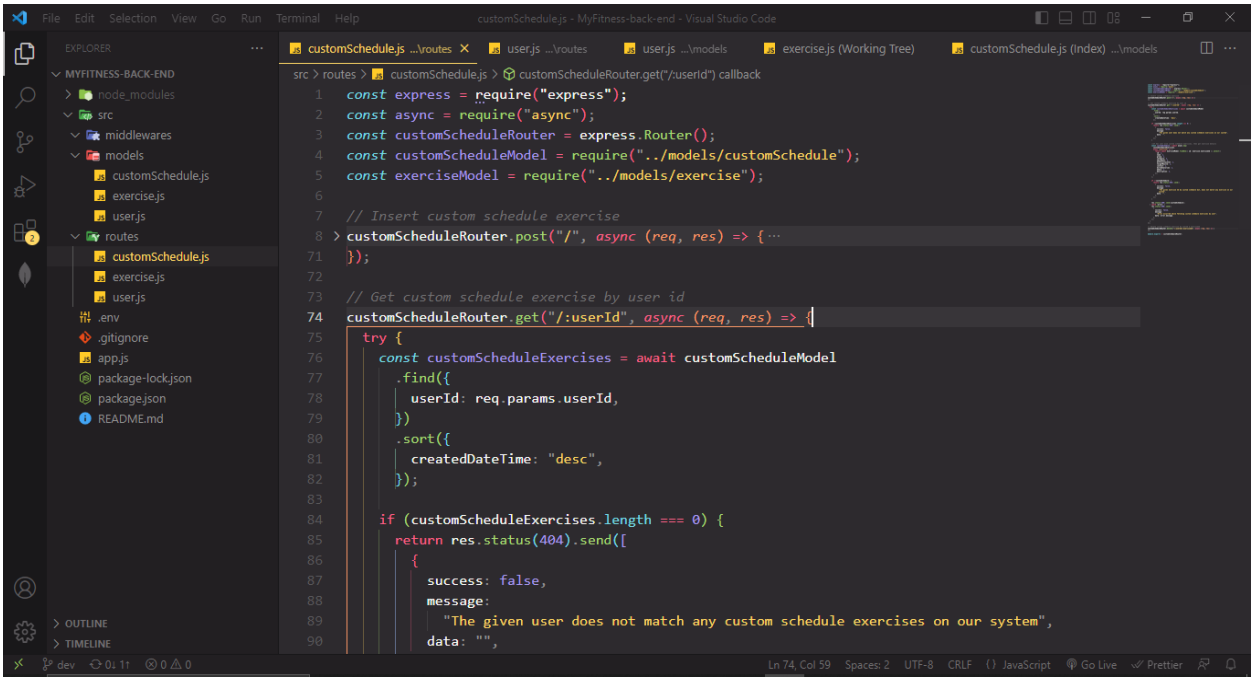
- The exercises in the user's custom schedule are displayed.
- User can remove an exercise by swiping a table cell to the left in the exercise list.



Back-End Development

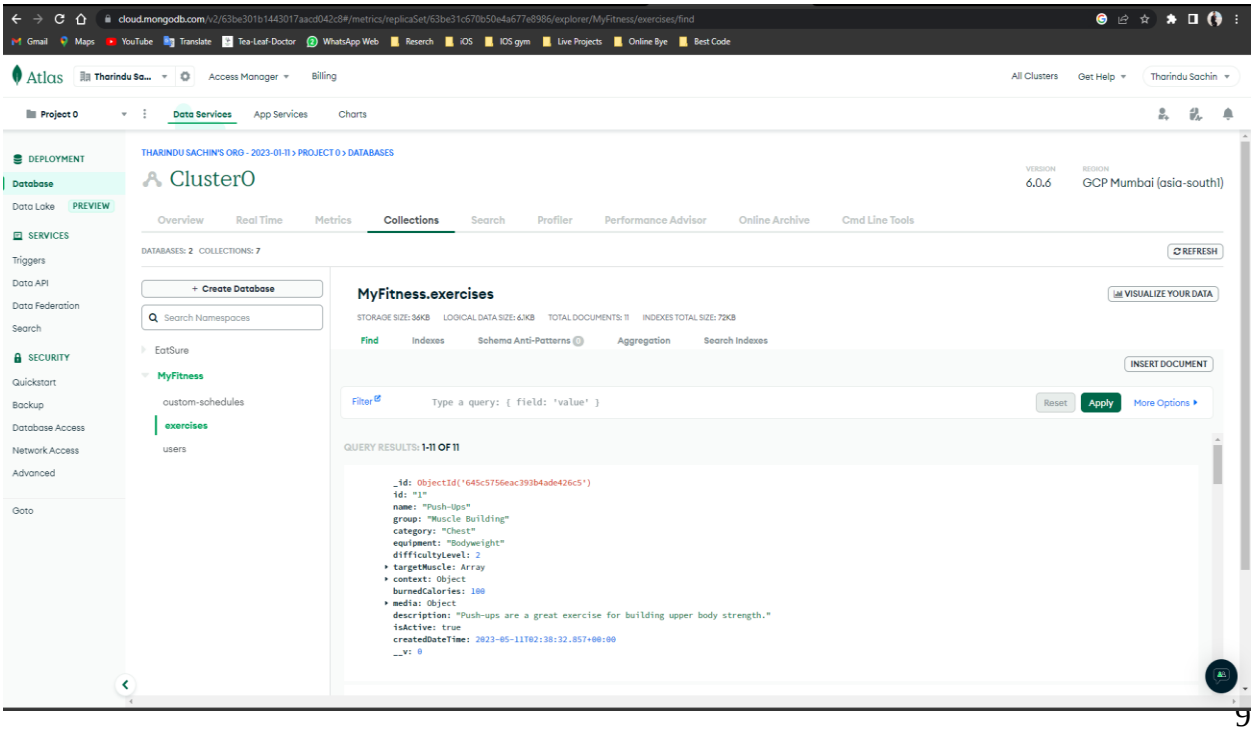
1) Web API Project

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



2) Database

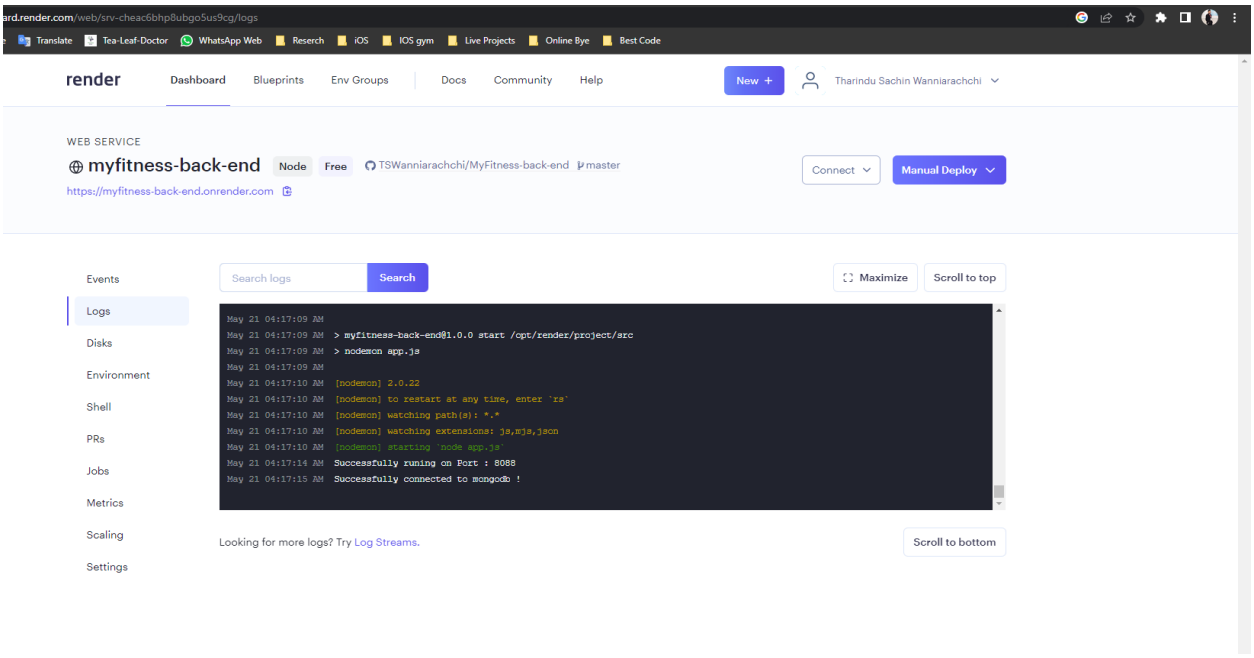
The MongoDB Atlas cloud database was used as the backend database for the project.



Other Development Tools

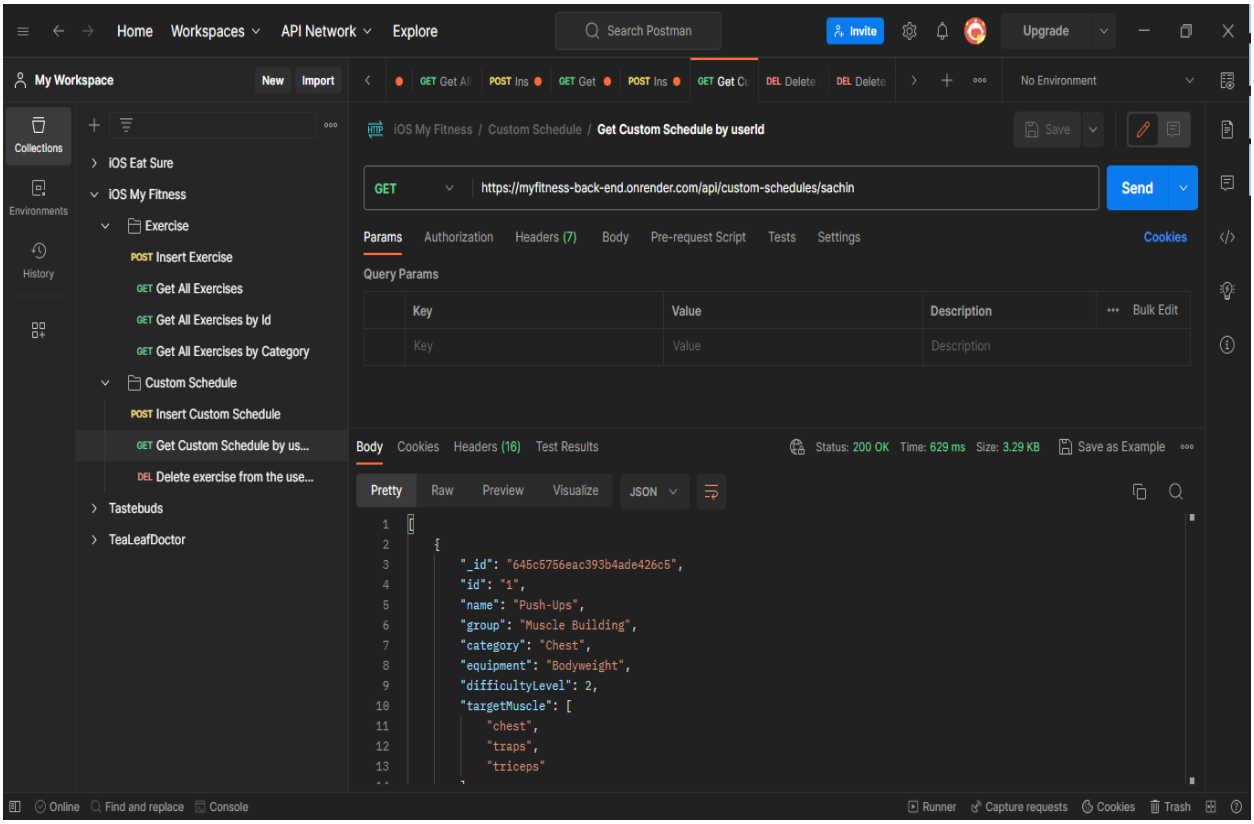
1) Render

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



2) Postman

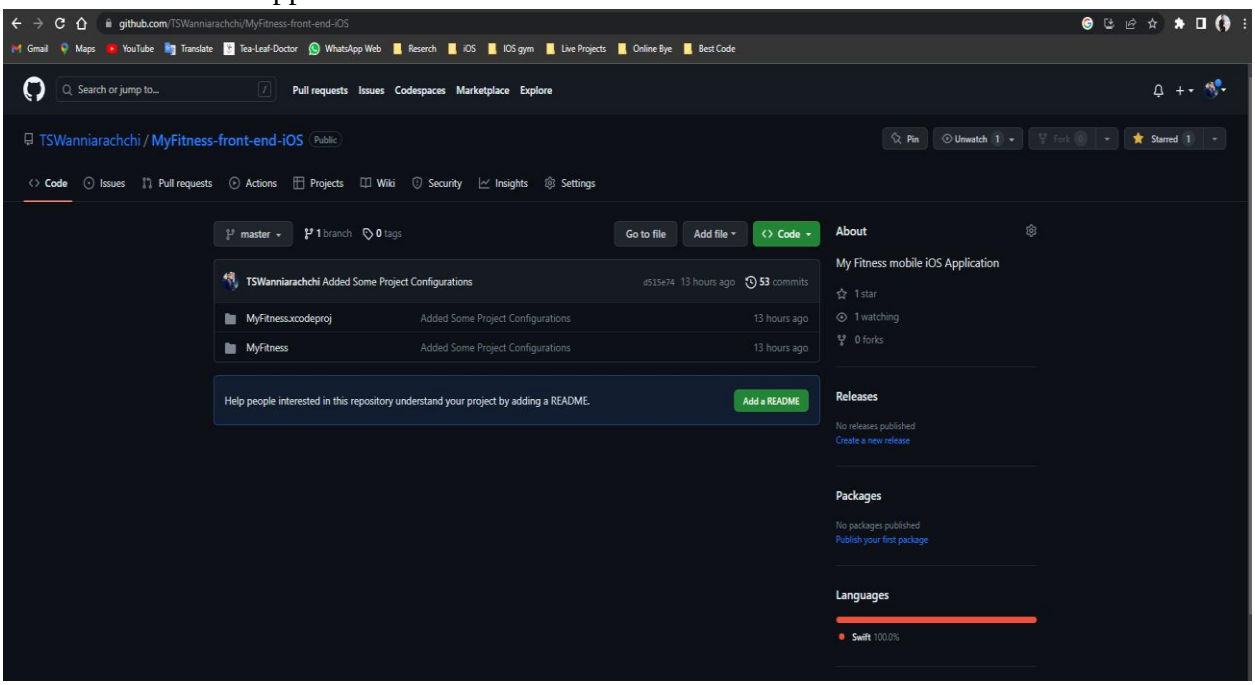
Postman was used to check the API requests



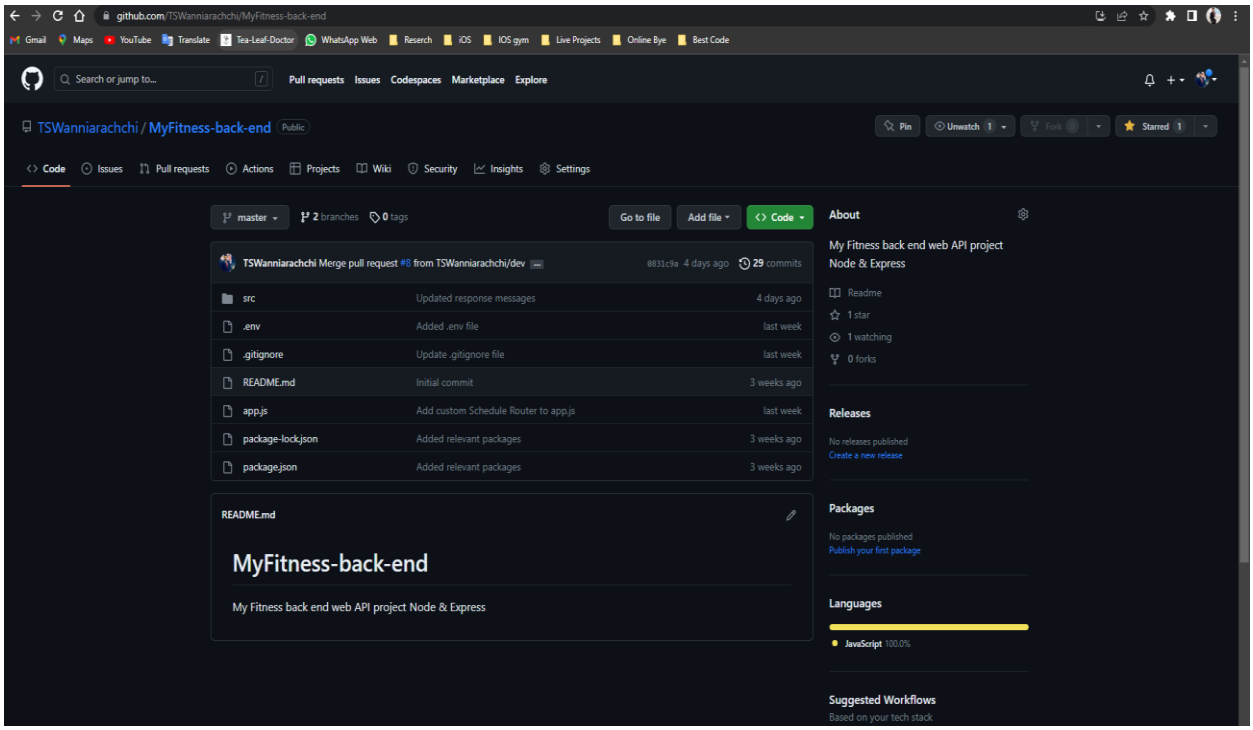
3) Source Control

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

i iOS mobile application



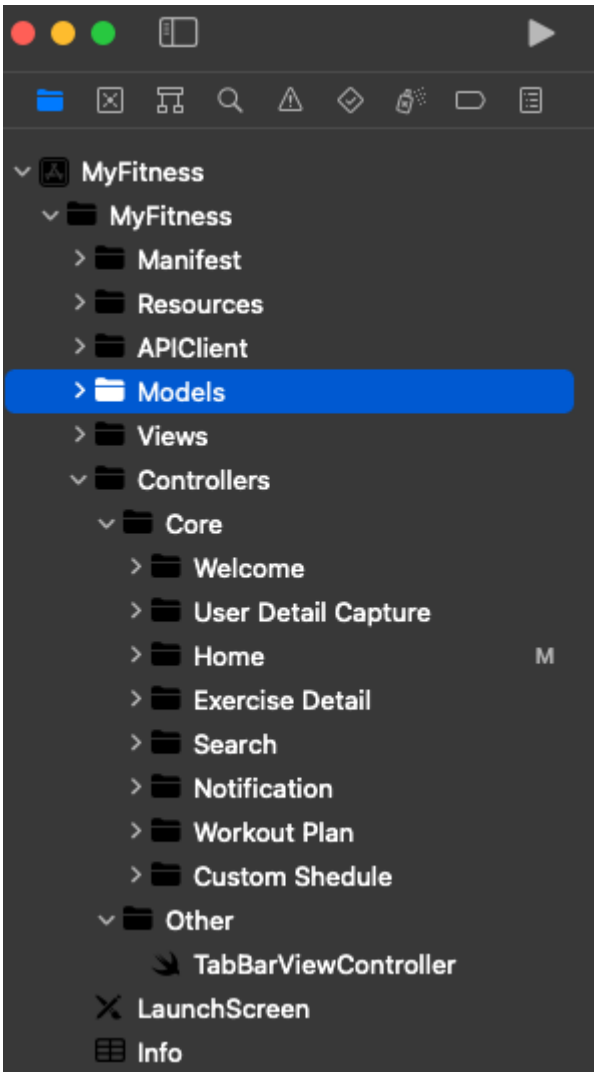
ii Node.js web API project



Using Best Practices

1) MVC Architecture

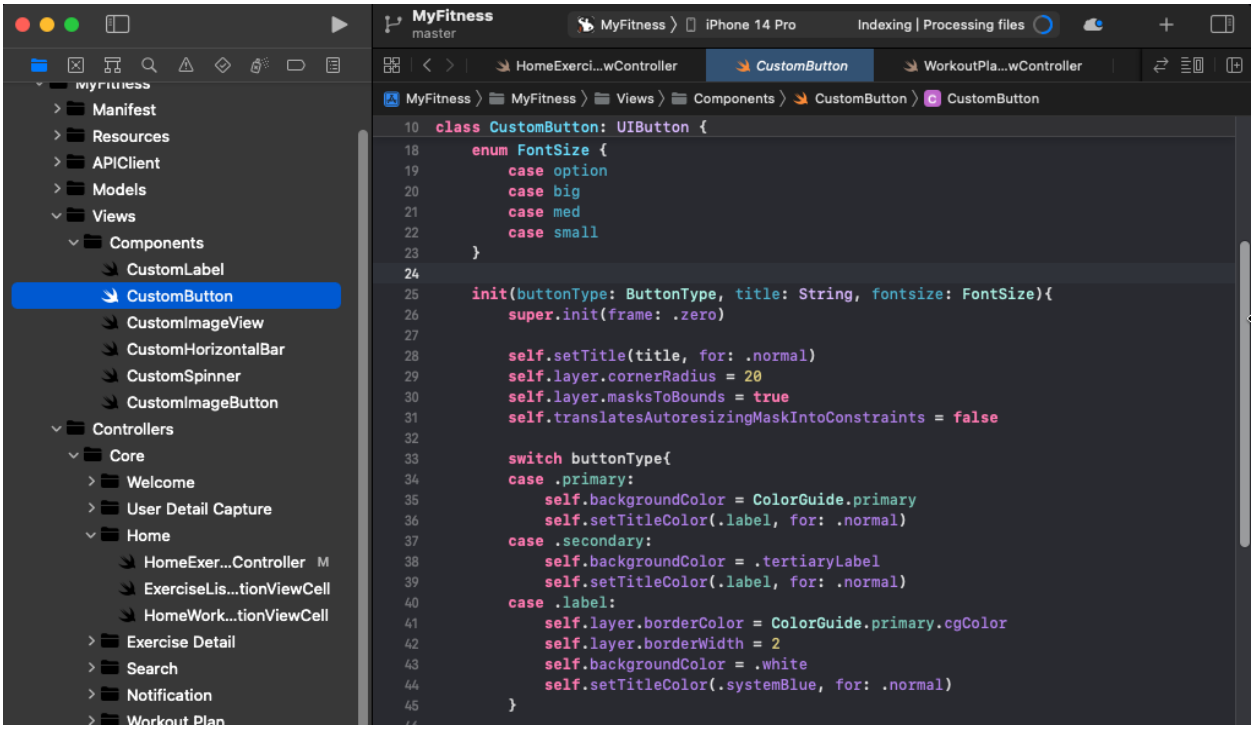
MVC architecture was used to create this application. It helps organize code properly by separating the view (user interface), models (database), and containing logic in controllers.

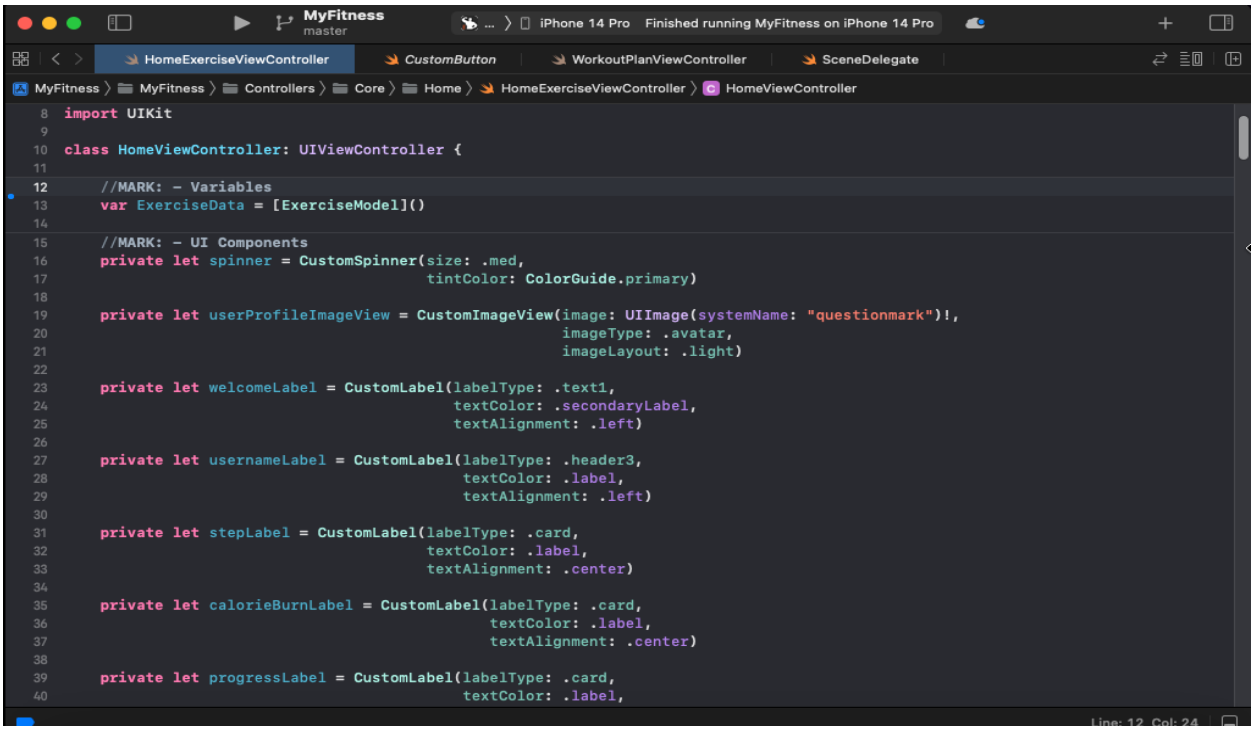


2) Reusable Custom UI Component

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

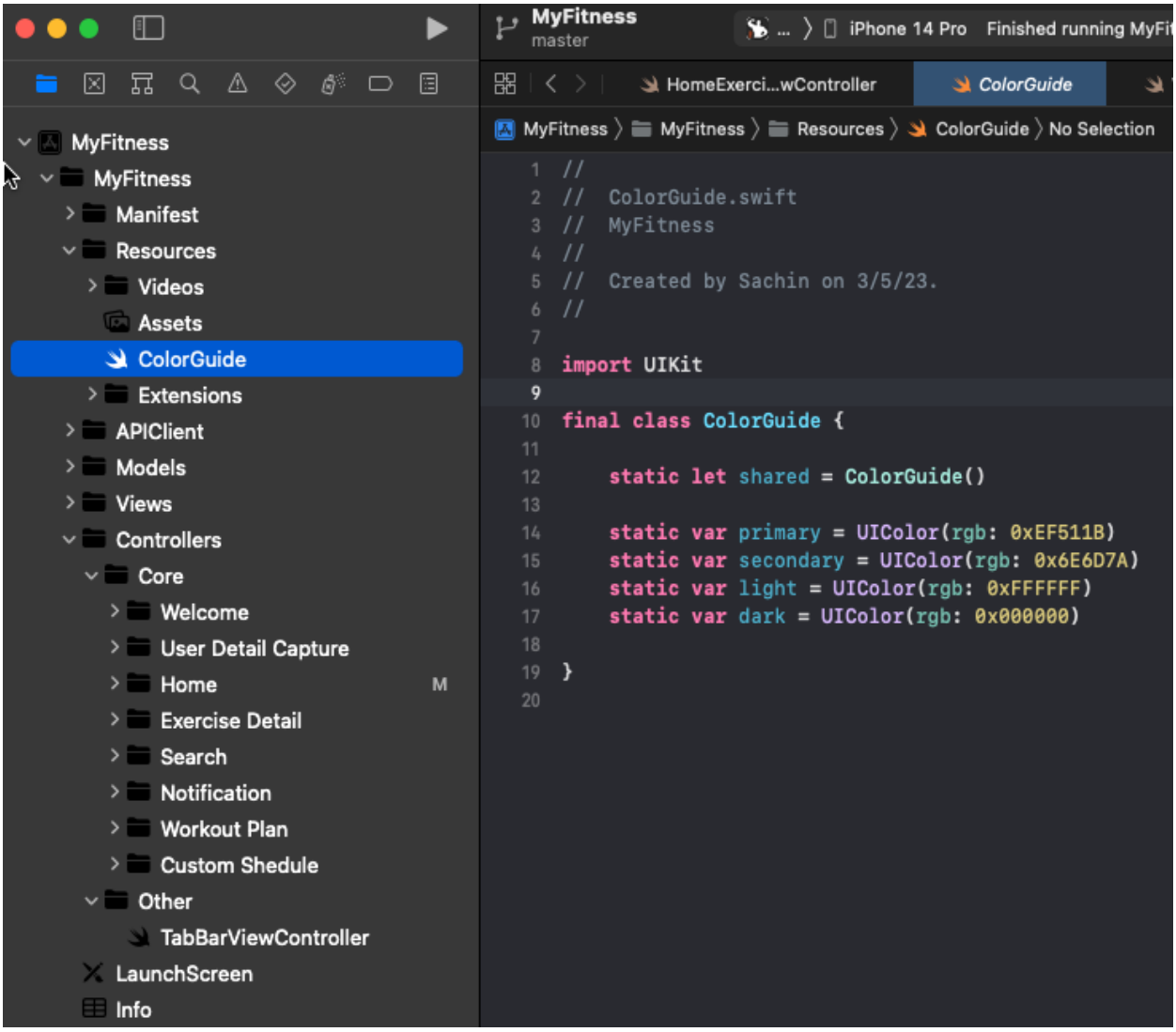
Below is the reusable UI Component and how it is used in a view controller.





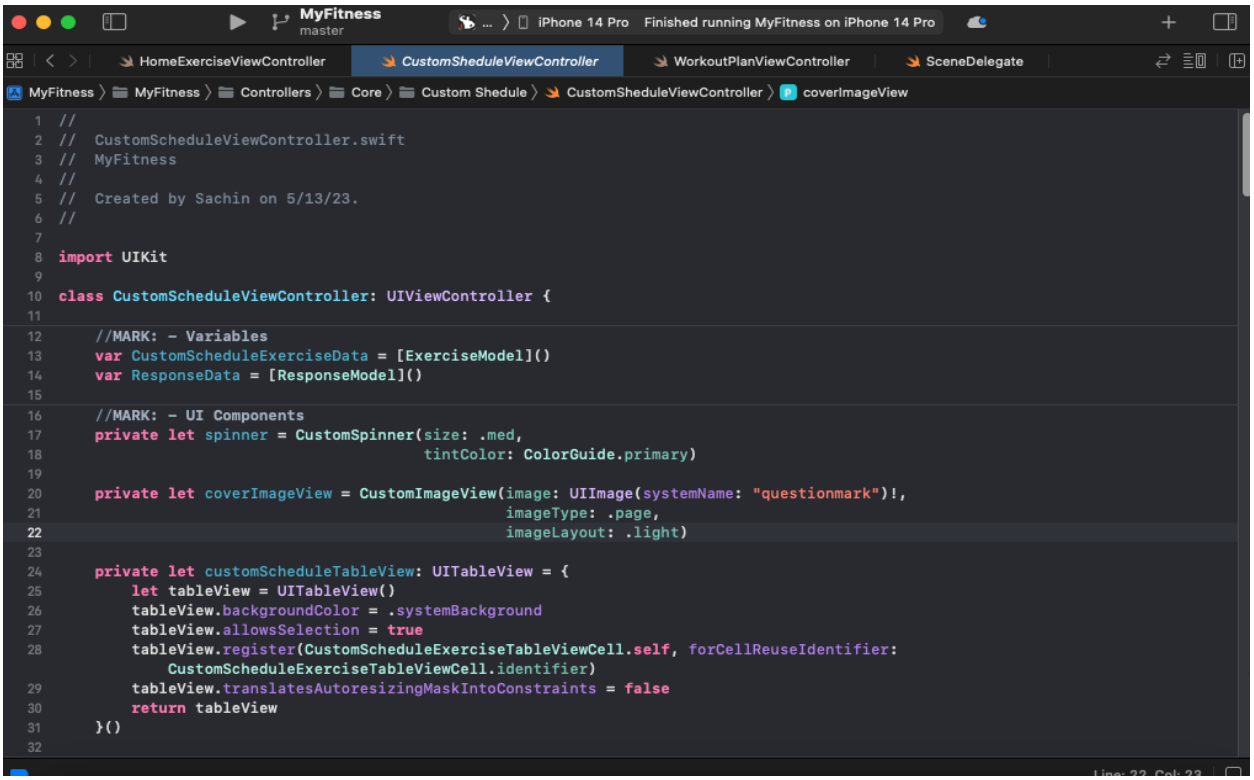
3) App Theme in Separate File

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

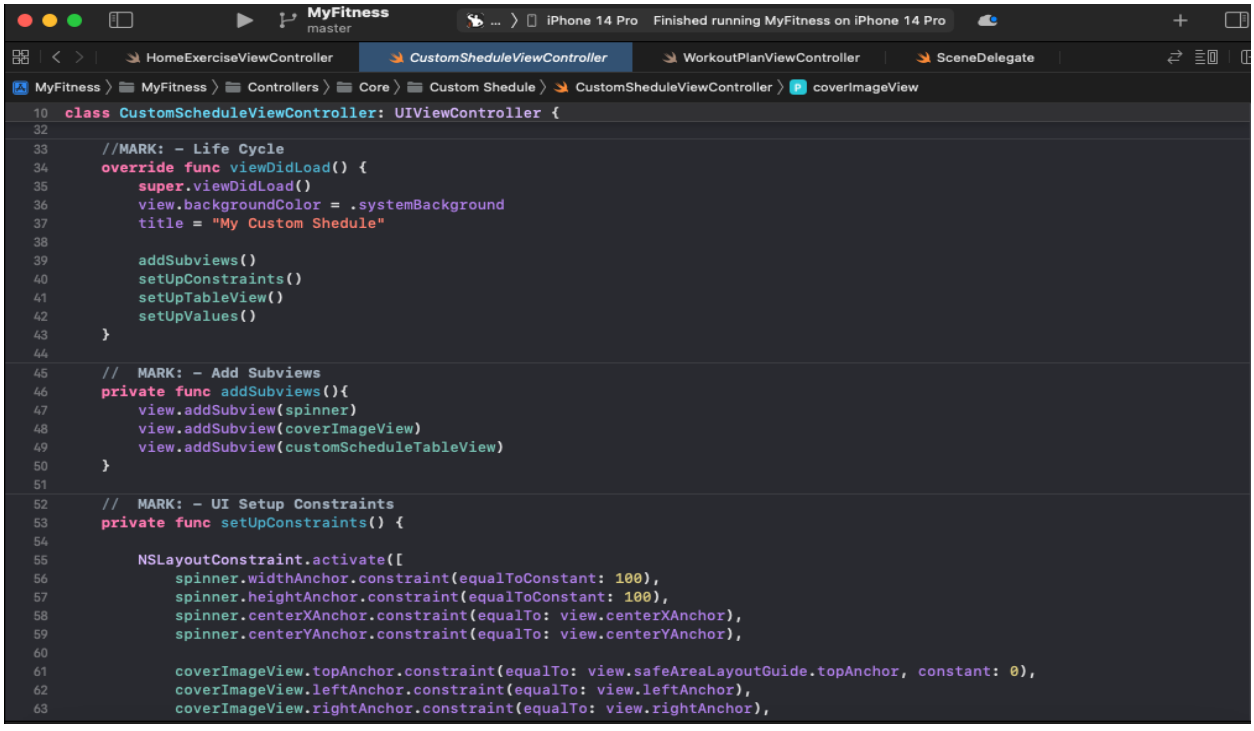


4) Manageable Code Structure When Design Interfaces

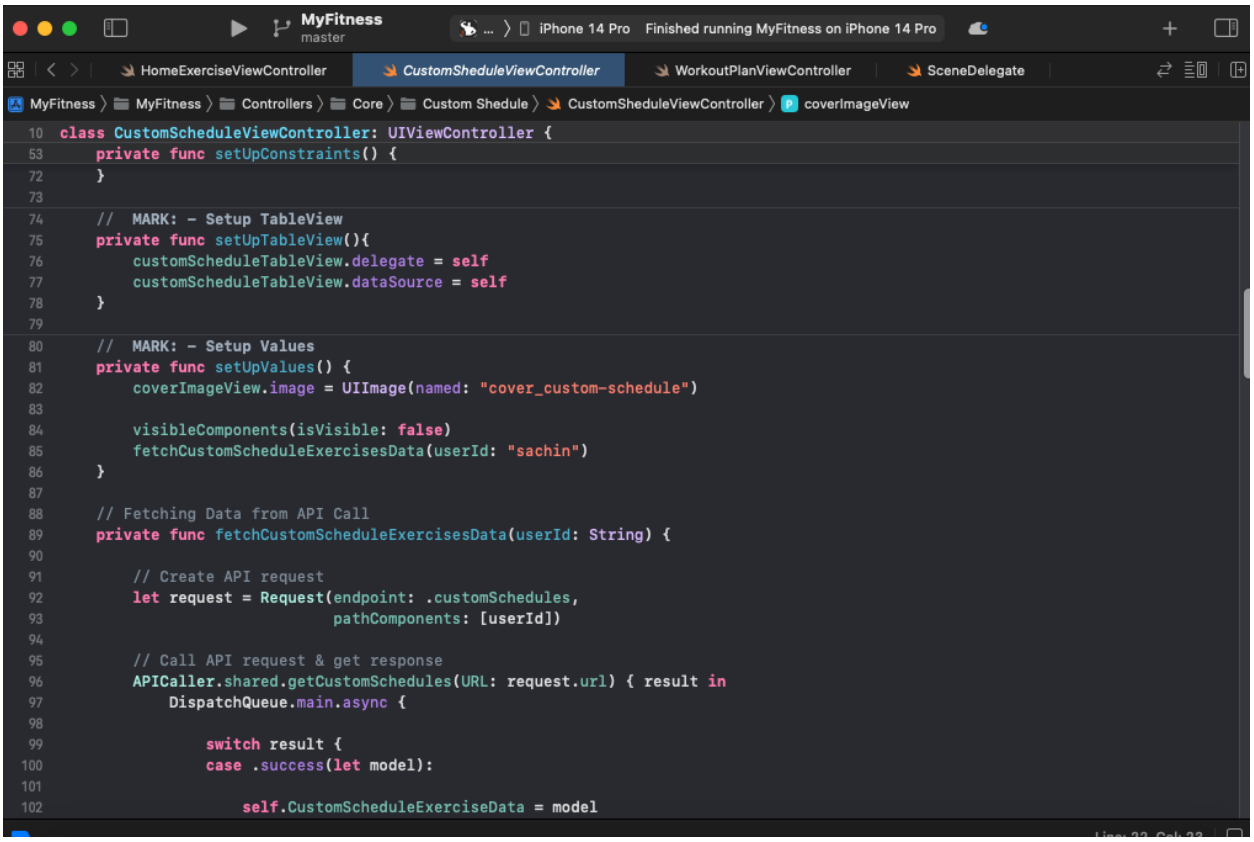
When designing the user interface, created a separate function as shown below and call the viewDidLoad() function. It made the code very easy to maintain. As a best code practice, the tasks performed by the relevant function were commented on.



```swift
//
//  CustomScheduleViewController.swift
//  MyFitness
//
//  Created by Sachin on 5/13/23.
//

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)

    private let coverImageView = CustomImageView(image: UIImage(systemName: "questionmark")!,
                                                 imageType: .page,
                                                 imageLayout: .light)

    private let customScheduleTableView: UITableView = {
        let tableView = UITableView()
        tableView.backgroundColor = .systemBackground
        tableView.allowsSelection = true
        tableView.register(CustomScheduleExerciseTableViewCell.self, forCellReuseIdentifier:
            CustomScheduleExerciseTableViewCell.identifier)
        tableView.translatesAutoresizingMaskIntoConstraints = false
        return tableView
    }()
```

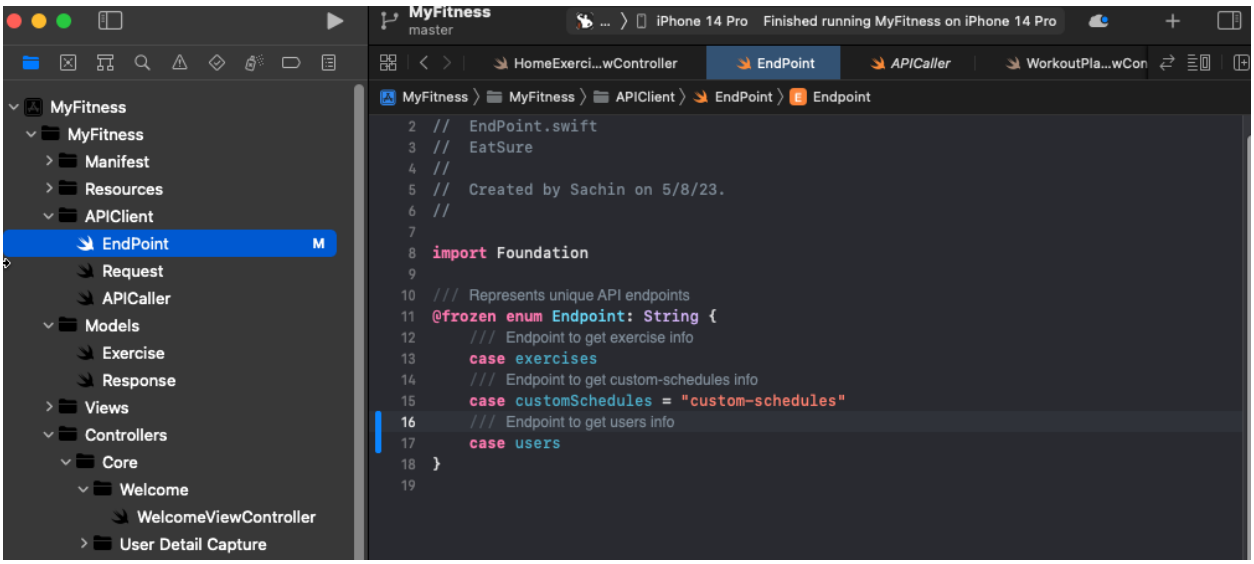


```swift
class CustomScheduleViewController: UIViewController {

    //MARK: - Life Cycle
    override func viewDidLoad() {
        super.viewDidLoad()
        view.backgroundColor = .systemBackground
        title = "My Custom Shedule"

        addSubviews()
        setUpConstraints()
        setUpTableView()
        setUpValues()
    }

    //  MARK: - Add Subviews
    private func addSubviews(){
        view.addSubview(spinner)
        view.addSubview(coverImageView)
        view.addSubview(customScheduleTableView)
    }

    //  MARK: - UI Setup Constraints
    private func setUpConstraints() {

        NSLayoutConstraint.activate([
            spinner.widthAnchor.constraint(equalToConstant: 100),
            spinner.heightAnchor.constraint(equalToConstant: 100),
            spinner.centerXAnchor.constraint(equalTo: view.centerXAnchor),
            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),
```

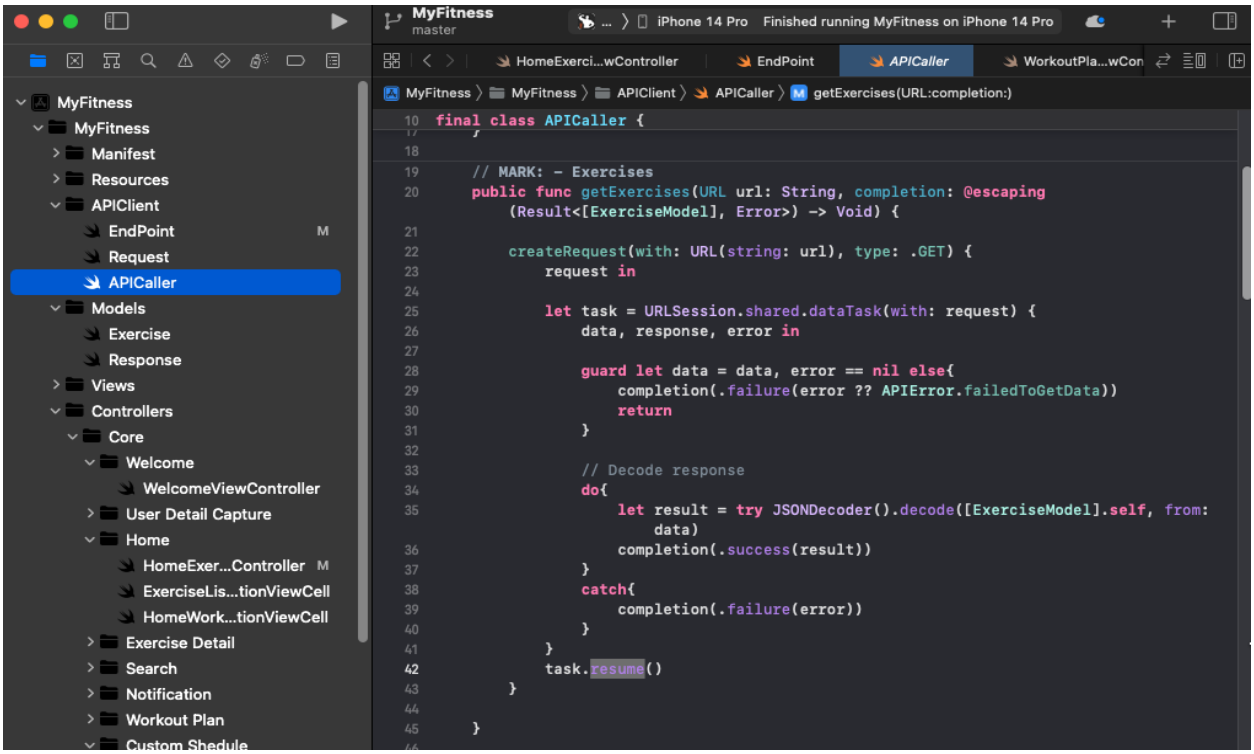


5) Created Separated Files to Manage API Services and Maintain Errors

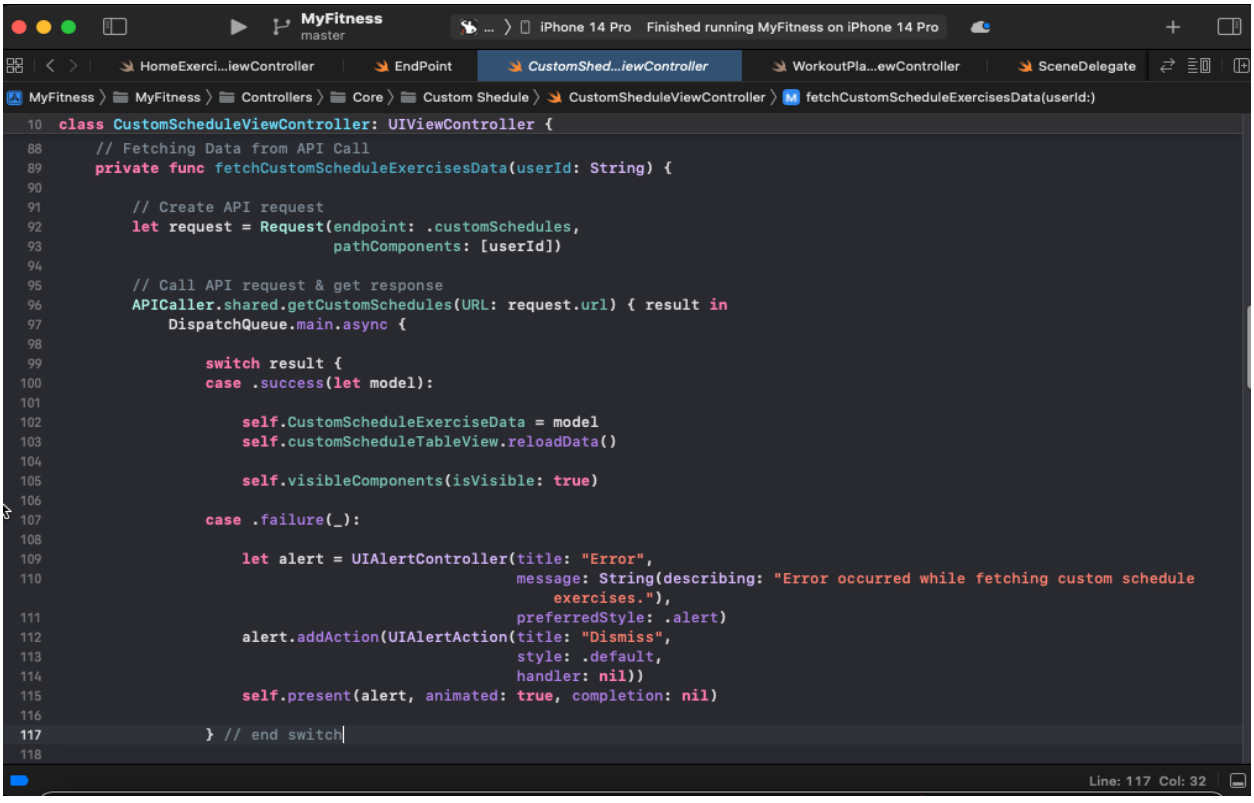
i Store End-Points



ii Maintain API Request in separate file



iii Fetching data from API call and error handlng



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-DgwFCV1o4/> [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].