OccSpec mock task 2, creating the solution.

Table of Contents

[Introduction 1](#_Toc94796600)

[Creation of the website. 2](#_Toc94796601)

[Programming the solution. 3](#_Toc94796602)

[Website design 3](#_Toc94796603)

[Pseudocode 10](#_Toc94796604)

[Model 10](#_Toc94796605)

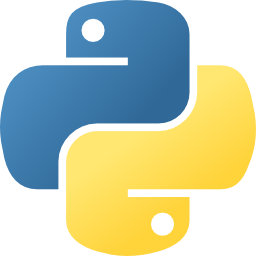
[Controller 10](#_Toc94796606)

[Login page 12](#_Toc94796607)

[References 13](#_Toc94796608)

# Introduction

In this document I will be: creating a digital solution for the client, testing the digital solution, documenting the digital solution and trying to improve upon it every time. I was going to use java as the back-end however tomcat is not installed on the computers so doing a webserver would prove very difficult, so I have decided to change it to python instead.

I will be using Python as my back-end as I am somewhat comfortable with it and can be used for a server. It is easy to use and good large numbers, but it is slow. I will use IDLE for my IDE.

Icon

Description automatically generated

I will use VSC or visual studio code to create the front-end as it supports many languages, but I will be using it for JavaScript. I am not fully well versed, but it should be easy as it links with HTML and that relativity easy to understand. I will also use CSS as it can be very useful in making HTML a lot more manageable to use. Overall these languages should work well together and should produce a good system.

The client has asked for me to create a digital solution that can:

* provide information and advice about fitness training and healthy living
* provide access to digital content to support customers with their training and healthy lifestyle
* encourage existing customers to use more of the services provided by [] Fitness.

The outlines that client has given me can be used to create a website that can display this information, and we will use python as the server to hold the information. The best place to start would be creating the HTML for the website and then moving onto python to create the requests and functions.

# Creation of the website.

I will start by making a basic looking website that will display:

* customer information.
  + name.
  + email.
  + telephone.
  + Extra feature: favourite workout.
  + Extra feature: favourite food/drink.
  + Profile pic if they want.
* Images.
  + [] icon.
  + User or people doing workout.
* Contact section.
  + Email for [] if client needs help.
  + Telephone for help line.
* Healthy lifestyle information.
  + Balanced diet information.
  + How to eat healthy.
  + What to avoid if you have medical conditions e.g., diabetic, allergic, heart problems.
* Fitness training information.
  + Best fitness plan for you.
  + Extra fitness options for people with disabilities.
* Have a system that benefits customers to use their services more. potential options:
  + Loyalty system.
  + Get updates a week earlier.
  + Discount on [] products.

# Programming the solution.

## Website design

I will start with designing the website, as well as programming it. the base of it will be HTML with JavaScript included. The first thing to do is to create a HTML file and type <!DOCTYPE html> this shows what version we are using and the browser to recognise it.

The few lines we have will produce a website with the title “Welcome to [] Industry” it very simple and not much. We use “<html lang = “en>” to specify the language; in this case it is English. We use “<meta charset = ‘UTF-8’” because it covers all character and symbols in the world (different languages).

The code I have written will produce a:

* Header; welcoming the user.
* Paragraph; tells the user what to do.
* Page break; puts a space between sections
* Form; creates a box asking for a Username.
* Form; creates a box asking for a Password.

I have decided to change the website, instead of logging on the website would display normal features like diets, fitness exercises, contact section and more. but for more features like emailing, premium features buying products, that would be done by signing in. I have changed some of the code around and added a CSS file to give it more character and depth.

Graphical user interface, text, application

Description automatically generated

A simple look, this will change overtime as we develop more to this website. I will add colour and some images to make It look more professional and cleaner. Objects may move around to be more in line with each other. What should happen when the code is finished is the login details get checked and if they are correct, it opens the users page if not it returns saying the details are incorrect. This would happen by sending a request to the server to check that details are correct if they are it opens the main window by returning their data to the webpage. Any process or feature will be requested from the server.

I have added some code to the CSS file it now saves colour, images, font size, font position, animations, margins, and more. I have tried many backgrounds and think I have found one that looks nice and should fit the website.

I have decided to go with a brick theme as it looks nice but might change the blue to a brick orange.

This is what the code looks like inside of the CSS file.

Inside I have all the features that help the website look better. I have also saved images in here as it easier to use than inside if html. I’m going to add more information starting with diet plans such as: good balanced meals, what to eat during the day, what best beverages to have and more. I will add images as well and maybe some gifs if they fit the website.

I have decided to change the website look and feel completely as it does not suite what I was going for. I have changed the background to a nice black and have changed the blue bar to a dark red. These colours should contrast of each other and make it have a better feel.

As you can see it looks a lot nicer than before, the colours play of each other well and give it a much better atmosphere. I will add in the top right a drop-down menu that will display:

* Log in; log into their account for extra features.
* Sign up; being able to sign up to the company.
* Contact us; a away for the user contact us for any issues.

This will all be hid behind a drop-down menu and when the user clicks on one of these buttons it will send them to another part of the page to do those actions. I have also added a gif for a funny gag and added some pictures for diet foods they might want to have in their fitness grind.

I need to make it so the pictures are centred with the text to make it look correct and proper but I am having issues doing this.

This is all the text above the pictures describing about what Toku is and what they aim to do. I have added pictures and had issues where they would not be centred in the middle of the page. This caused me a lot of frustration and had to try many different solutions from trying to make the CSS set the picture centred to making the margin do it. in the end I was able to make the images centred by making it in line with the pictures. This fixed it and saved a lot of effort for me.

I will not implement a drop box for mobile devices as well as to make the website have more dynamic pictures. I firsts started with what I want to put in the drop box, but I went with these three options. Contact us, login in and sign up. These will be display inside the drop-down box/ menu. I have many issues with creating this drop-down menu. I used w3schools code and changed some bits here and there, but it did not fully work and created many issues as well like causing the screen to become minimized, text in wrong places, not creating a drop-down box and making a line of text instead and much more. This is the website I used (w3schools, n.d.) for the menu. a lot of the code is from there but I will be changing it to fit our website more.

As you scroll down more, I will probably add what fitness types are good for you and what best to do for you. I will add a login system that will be used to get there, emails, username, password, and other information. I will code it in python, as I can use3 flash to send requests for the server and should be relativity easy.

This is what the website looks like with the drop-down menu.

This will not help people with pc very much but will be very beneficial for mobile devices as they can now navigate much easier and can fit more on their screen. I might change the colour but for now it is okay. I have started to code the server. This will hold all the user data and be used to take information and sent to the user as well as retrieve from them as well. We will use python for it as I have decent knowledge of understanding it. I will use GET and POST request to give some security and efficiency to the code. We will house all the user data inside of a CSV file, when we ask for the user data we will ask for: ID number, Username, Password. This will be used to give us easier understanding of what user it is, as well as giving them and us more security to protect their data. Once the website is up and running, we will encrypt the data to give extra security as all the data is saved on one file.

## Pseudocode

Creating the server that holds all the data and sends the request to website should be somewhat easy. I am going to do some simple pseudocode to get understating where I need to start and what needs to be added when I have written the start of it. I will code it in python as I have some knowledge on how to do request. We will use two different files for the server. The model and controller are names of the files, and they will communicate together to send information to the website.

### Model

From flask import render\_template

Class Model:

def GetLandingPage(self):

return render\_template(‘Tasks2Code\_Website\_478905\_Clare\_J’)

def GetLoginPage(self):

return render\_template()

def GetLoginDetails(self):  
 return render\_template()

This should be enough to get me started on the model, I will not code a bit of the controller as well.

### Controller

from flask import Flask, request

from Model import \*

app = flask(\_name\_)

model = Model()

@app.route(“/”)

def LandingPage():

return model.GetLandingPage()

@app.route("/LoginDetails")

def LoginPage():

return model.GetLoginPage()

@app.route("/FetchUserData")

def LoadUserData():

return model.GetLoginDetails()

the pseudocode I have made should make it easier to program all of this. Will start by trying to get some simple feed back to website like changing the colour and then will work my way up to displaying user details from a CSV file. I have added the pseudocode lines and will now begin trying to update the website.

## Login page

Before we code more in python, we will need to add a login system. This will allow the user to input their credentials to access their account. This will link to our other website by the drop-down menu. When they click on the menu it will show 3 options, if they click the login menu it should open the login page and allow the user to enter their details. There details will be pulled from a CSV file I have created called ClientsData that will hold everything, it will then show their account. I have shown example below to make it clearer.

Click this icon to see options.

Click login to be taken to login page.

You will be taken to this page to enter your details.

Username and Password

Once this is done it should check the csv file to see if the details are correct. So far we only have one user inside of the csv file which is a test dummy to see that the file is being read by the software and can be shown to a human.

So far it out prints the users: ID, Username, Password, Email, and date of birth. When the user input their credentials, it will check the csv file to make sure it corrects. When I was creating the login page, I had to use something called Flex to centre the boxes and text on screen. This allows me to make the login page nicer and much easier to design. I got some code from (w3schools, n.d.) as I was not sure how to use it.

This code allows me to center the header and text area, this was useful as cut me a lot of time. The only code I have in html is buttons a text as nothing else is really needed as of the creating the product prototype. I have been able to get the python code to run in the controller, it shows this code when on boot up.

Only issue, is after booting it up and typing “LocalHost:5000” in the search bar on Firefox it throws and error saying it does not work. This is an issue I’m hoping to resolve quickly to save time for other parts of it and be able to get it up and running.

# References

w3schools. (n.d.). *How to*. Retrieved from Mobile navigation menu: https://www.w3schools.com/howto/howto\_js\_mobile\_navbar.asp

w3schools. (n.d.). *w3schools css*. Retrieved from CSS Flex Container: https://www.w3schools.com/css/css3\_flexbox\_container.asp