HSC 2024 Major Project:

Problem Definition:

This software aims to reduce the amount of excess time undertaken by employees who wish to extend their knowledge of fruits and vegetables at Harris Farm Markets. Currently, at HFM they are completely lacking a system in which workers can learn the fruits and vegetables present in the store. Problems arise when produce changes due to seasonal differences and when new hires are introduced to working at Harris farm markets. In the case of new workers, the nature of being new is stressful enough, and their lack of training systems contribute to the problem. My application would not only reduce the stress incurred by employees, but also generate more revenue for the company reducing the amount of wages paid during training time.

The process in which this is achieved is through a testing application that would be present on the work computers. This software allows users to take tests with twenty questions, within these questions a prompt is given showing a fruit, or vegetable. Users will then decide what they determine the prompt to be out of four multiple choice options. Scores will be saved to a database, and then users will be able to see their average score, when five or more tests are completed. Managers will be given notifications when a user is falling behind company standards, this standard can be changed depending on the user, and what time of year it is (due to the nature of fruits coming and going with the seasons).Due to the time constraints that are apart of the development of this project, this is the complete scope of the task, and it will not go beyond what has been previously stated.

As a developer I am extremely invested in this project due to my position as a worker at Harris Farm Markets, and I have been doing so for the past three years. I have observed this problem firsthand and the negatives associated with it, as you can see on the review page of Harris farm, this problem is detrimental to the store's reputation.

Anastasiia Galiulina
Local Guide · 22 reviews · 42 photos

★★☆☆☆ 2 months ago

Be sure to take your receipt and check prices at the cash register!
1. The store periodically has incorrect price tags in the store.
2. At the checkout, staff make a lot of mistakes. Every second purchase contains mistakes. Once they calculated my purchase to be $18 more than the actual one. This time I paid $7.49 for the ricotta instead of the advertised price of $3.29.
Otherwise, I note a good selection of fresh vegetables and fruits, and polite staff.

UPD: I got a refund.

4

Social And Ethical Issues:

When developing an application such as this, social and ethical considerations must be taken into account to ensure a reputable and streamlined integration process. The nature of my application raises several needs for inquiry when it comes to social issues.

The purpose of my app directly raises many social issues when it comes to the job security of employees within the company. Because my application streamlines the training process, it may lead to a loss in workload for human trainers, and because of the nature of wages and how they are given by the hour. The integration of my application could lead to the reduction of hours required for human trainers, or even complete loss of job. Which would not only damage the company's reputation for workers, but would also affect the broader community. Additionally, necessary training would have to be completed to get users up to speed with the software, as some users may find it digitally challenging to use the software effectively.

When storing data on users it is natural for concerns to be raised, my application stores usernames and passwords as well as the tests completed, and average scores of users. Due to the nature of legal laws within Australia, users will need to be informed of this monitoring, and if they wish can opt out of the system. To achieve this the application will feature a username of ‘Guest’ and password of ‘Guest’ so users can vindicate themselves from any monitoring and/or data breaches. Copyright is also an issue, and within the company fair use of intellectual property must be used, making sure no laws are breached.

Whenever developing an application, developers must take into consideration the needs for visually impaired and people who require certain features to be enabled. My application will try to meet the best practice standing when it comes to the inclusivity, ethicality and legality of my application.

Functional and Nonfunctional Requirements:

These are the requirements my application must have in order to achieve its problems definition.

Functional:

- The application must have logic compiling 20 unique fruits and vegetables into a multiple choice quiz.
- Users must be able to get a score out of twenty upon completion of this test, and later have an average.
- Managers must be able to see the performance of users, in relation to other users across the application.
- A login and registration page is required as users test data must be monitored in order for users and managers to track progress.
- Passwords must remain hashed in storing to increase security in the case of a data breach.

Non Functional:

- It needs to be clear when users get an answer wrong, indicating with a sound or colour change on a button.
- Images need to remain the same size, and fruits and vegetable images must be large enough for users to see them and point out general features.
- Managers must be able to see the data in a manner that is easily distinguishable and is not too complicated.
- Colour scheme must remain the same throughout, with best practice UI and UX principles remaining similar.
- The user average must be calculated as an integer, otherwise if it is created as a float large recurring decimals might cause the UI errors.

Diagrams:

Structure Chart:
https://viewer.diagrams.net/?tags=%7B%7D&title=Structure%20Chart.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1YsZzTD369w1BCVQ5sFmdDBdCNuDtvzwZ%26export%3Ddownload

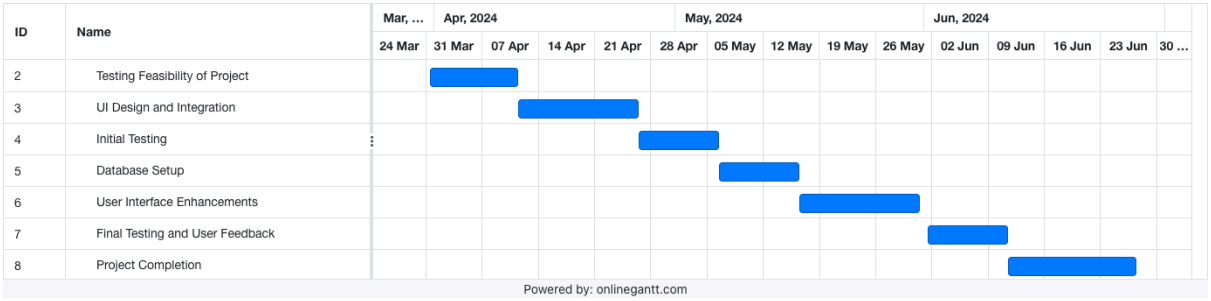
Algorithms:
https://viewer.diagrams.net/?tags=%7B%7D&title=Algorithims.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1aebeCgt9NG09RlXvkNO82KZ2jelzmXpV%26export%3Ddownload

Context Diagram:
https://viewer.diagrams.net/?tags=%7B%7D&title=Context%20diagram%20for%20HFM.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1soLtEBX-D9U0SK6YWsdtmiDAlszrxT32%26export%3Ddownload

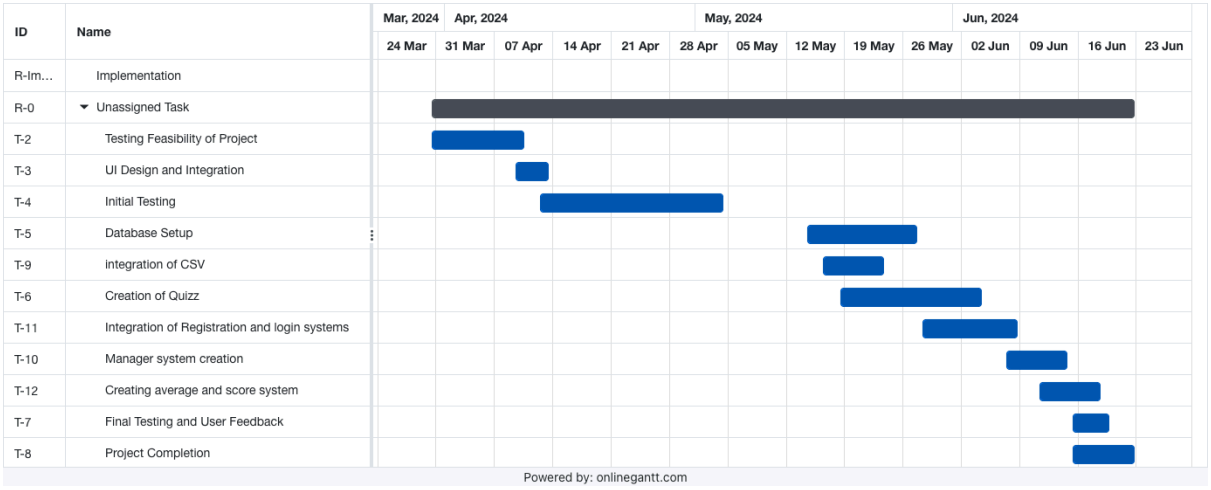
Data Flow Diagram:
https://viewer.diagrams.net/?tags=%7B%7D&title=DFD%20MJ.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1lzZyKVnwxs8l92VMPJ-vVZe9_M61W5RJ%26export%3Ddownload

Storyboard:
https://viewer.diagrams.net/?tags=%7B%7D&title=MJ%20Storyboard.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D1L9cOLPO9BqaqynBHhDLZBH8Su34H_KFT%26export%3Ddownload

Original Gantt



Later introduced Gantt



Reflection:

Overall, the time management of this project could have been better, however due to the busy schedule it was difficult to dedicate enough time on the project. The overestimation and underestimation of tasks is evident within both my Gantts, evidencing my lack of knowledge in regards to project management. It is specifically difficult when it comes to coding projects, as often, as in my experience, you run into issues that prolonged the bug fixing and thus the length of development. Specifically within this project, because I lack knowledge on some of the systems it took longer to create the project by the original deadline. Overall, to further my project management skills it would be beneficial to take into account people's ability, and estimate how long it would take to develop something in account of these abilities.

Development log:

This is a development log of the different things being completed throughout each week of the development of my project. Refer to the github progress/commits to get an outlook on the code completed. Monday is the start of the week, not sunday.

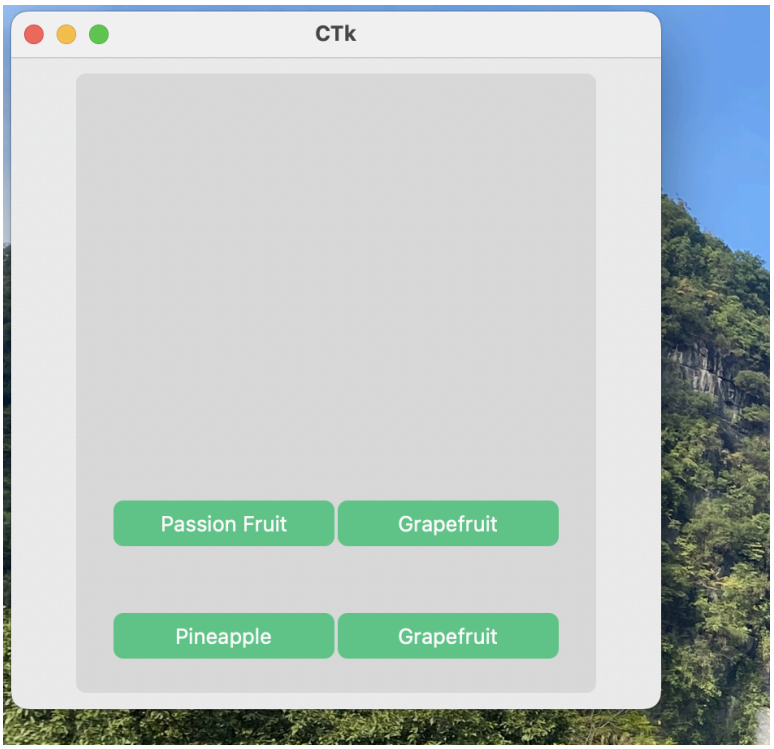
Date:	Saturday 30th of March
Week:	Week One
Objectives:	I have decided to create some terminal apps that achieve the basic purpose of my app, basically a test if the project is

	achievable. The application will run in the command line and users will input the answer to what they think the question is.
Snippet of code if any available:	**Keep in mind there is an array above that contains all the fruits and vegetables, however it was too large to contain in this table.** `class fruit_and_veg():` `  def __init__(self, score, process):` `  self.random_int = random.randint(0,76)` `  self.score = score` `  self.process = process` `  def randomfv(self):` `  random_fruit_or_veg = fruits_and_vegetables[self.random_int]` `  print("what is", random_fruit_or_veg)` `  answer = input('')` `  if answer.lower() == random_fruit_or_veg.lower():` `  self.score +=1` `  print("your score is", self.score)` `  else:` `  self.score -=1` `  print('not correct', self.score, 'is your score')` `  if self.score == 20:` `  self.process = False` `  print('well done, you have completed the test with a score of', self.score,'/20')` `is_process_running = True` `current_score = 0` `while is_process_running == True:` `  random_fruit_and_veg = fruit_and_veg(current_score, is_process_running)` `  random_fruit_and_veg.randomfv()`

	`current_score = random_fruit_and_veg.score`
Problems and solutions:	Learning object oriented code was difficult at first, however I have learned the ropes of the basic concepts. I plan on using it in my major project as it makes the code more manageable.

Date:	Tue 9 April
Week:	Week Three
Objectives:	https://www.youtube.com/watch?v=IJ-iVnN09-8&ab_channel=Atlas I wanted to come to terms with the UI system that is custom tkinter, and I have implemented it into the code. I want four buttons in the latter half of the screen that will turn into the four unique answers for the prompt that will come soon. https://github.com/IggyMeagher/major-project/commit/c6337f8a0ffc5e6e1710a59830519a696ed8b72e Refer to the above link to see the code currently.
Problems and Solutions.	I have never seen a pack, place or even grid system that is like this before, and I am finding it pretty difficult to get the UI to work properly. Making the four buttons look good is very difficult and having them evenly spaced on the screen is harder.

Date:	Wed 10 April
Week:	Week Three
Objectives:	I want to figure out how the grid and pack system work still, so the four multiple choice buttons can look good within the app. I want random fruits and vegetables to show up when any of the four buttons are pressed too.

	https://github.com/IggyMeagher/major-project/commit/c6337f8a0ffc5e6e1710a59830519a696ed8b72e Click this link to view the code: Click this link, to view the array. As you can see, the array is very long, and doesn't really facilitate clean code. I will probably, later down the line, change this to a .txt file, as in preparation for ~300 or so fruit and vegetables being added, and having ~300 fruit and veg in an array would make for clunky code, and unnecessary space being taken up.
Problems and Solutions.	Currently this is what my application looks like:  I have been having trouble organising the buttons seen above, into a grid layout. As you can see the Y axis positioning for the buttons is very far apart, I would like these buttons to be closer, for the UI to look more presentable. Previously, the coding projects I have undertaken involving Ui have been through the API Gooey Pie, this api has a great method for aligning widgets along a grid, this is different within tkinter, as there are multiple ways to position widgets. I have divulged to use both a grid system, as well as a PADY and PADX system which is evident within the Github commit.

	This system is completely new to me, and it is quite a confusing process. I think to solve this issue, the padding will put the padding to 0 on the vertical, and add more grid, rows and cols, to the application to allow for precise placement.

Date:	Friday 12 of April .
Week:	Week Three
Objectives:	Doesn't actually work, so I have decided to introduce the *sample* function found within the random API that comes with python. This ensures a unique selection every time.
Snippet of Code:	`def checkingiflastnumberinarray(self): #There was a list out of range issue, so i created this function.` `for i in range(len(self.CumulatedNums) -1): #its probably over engineered, but it check if the last num of the array and makes sure the arr doesnt go out of range` `if i == 4 and self.CumulatedNums[i] > self.CumulatedNums[i-1] and self.last_num_of_array == True:` `self.last_num_of_array = False #returning this, so the next for loop can run without any issues.` `def MakingSureNoRepeatedLabelsForButtons(self): #checks if there are any repeated values, so they dont show up on the buttons` `for i in range(len(self.CumulatedNums) -1):` `if self.CumulatedNums[i] == self.CumulatedNums[i+1] and self.last_num_of_array == False:` `self.CumulatedNums.pop(i) #if repeated number is evident, it gets removed` `FruitsAndVegetables.pop(i) #it also gets removed from the other array, so it cant be repeated.` `i = i+1 #process is repeated.` `print(self.CumulatedNums)` `if len(self.CumulatedNums) <4: #checks if number is missing` `self.CumulatedNums.append(randint(0,20)) #a new number is made, so the removed one can be replaced`

	`` else: i = i+1 #contitues on if no number is repeated ```
Problems and Solutions.	It was frustrating having to debug my code, so tomorrow I will implement this sampling system

Date:	Sunday 14th of April
Week:	Week Four
Objectives:	Trying to create a function that whenever a multiple choice button is pressed it gives the user another question, and I added a label for testing that shows the correct answer. I made it so the correct answer is a different button every time.
Problems and Solutions.	No problems found, smooth integration of the function. https://github.com/IggyMeagher/major-project/blob/main/App.py Above link details the changes. I added comments as well, ensuring readability and later I will change the variable names so they suit their purpose better.

Date:	Wednesday 2nd of May.
Week:	Week Five (of development)
Objectives:	took a break from coding the project over the holidays, with school commencing I have decided to pick up the pace of development, as the project is due in the latter half of the term. So today's objective is to remove the array, therefore increasing the readability of the code.

Problems and Solutions.	Implemented a .txt file containing all the fruits and vegetables, instead of a python list (array) in the beginning of the code. This implementation increases the readability of the code, as well as making new additions to the fruit and veg bank easier. Have a look at the commit for yesterday as I forgot to commit all the code on the day it was done. No problems, I had to get some guidance on how to open TXT files as I forgot.

Date:	Thursday 2nd of May
Week:	Week Five (of development)
Objectives:	took a break from coding the project over the holidays, with school commencing I have decided to pick up the pace of development, as the project is due in the latter half of the term. So today's objective is to remove the array, therefore increasing the readability of the code.
Problems and Solutions.	Implemented a .txt file containing all the fruits and vegetables, instead of a python list (array) in the beginning of the code. This implementation increases the readability of the code, as well as making new additions to the fruit and veg bank easier. Have a look at the commit for yesterday as I forgot to commit all the code on the day it was done. No problems, I had to get some guidance on how to open TXT files as I forgot. https://github.com/IggyMeagher/major-project/commit/a7d0b11f7644ad377399bc393c9cdcafdecaae7c Above is the commit.

Date:	Tuesday 7th of May
Week:	Week 6
Objectives:	Currently there are a lot of bugs within my program, and I wish to remove these bugs if possible

	Another optional objective would be to have a login page where user data is stored, and since I have already created one it should be relatively easy.
Problems and Solutions.	Bug fixing has been a huge issue, with the accumulated nums not working as intended. I found out that it was actually appending to an array within an array. So I fixed that promptly by removing the cumulated nums array. I realised the scoring system did not actually pinpoint a right answer so I fixed that by changing the algorithm seen here: `def ListeningIfCorrect(self, clicked_button):` `self.correct = False #orgininally starts off as false` `if clicked_button._text == self.label._text: #checking if the answer is right, a bit buggy for some reason it doesnt detect the paramater` `self.correct = True #if the correct label is correct` `self.score = self.score +1 #changes the score accordingly` `self.update_questions() #gives the user new questions` `else:` `clicked_button.configure(text='✖') #informs the user if the answer is wrong, and makes them continue untill it is correct` `def update_questions(self):` `self.CumulatedNums = sample(FruitsAndVegetables, 4) #generating new questions` `self.label.configure(text=self.CumulatedNums[randint(0, 3)]) #generating new answer` `self.surveybutton1.configure(text=self.CumulatedNums[0])` `self.surveybutton2.configure(text=self.CumulatedNums[1])` `self.surveybutton3.configure(text=self.CumulatedNums[2])`

	`self.surveybutton4.configure(text=self.CumulatedNums[3])` Now it checks the label that is clicked, rather than all the labels at once.

Date:	Saturday 11th of May
Week:	Week 6
Objectives:	I want to use the full functionality of object oriented code, removing the extent of repeated code within my application. I will achieve this by creating a masterclass that is the Page class, which will make it easier to create new pages and remove the need to create new frames, geometry ect. This will further set up cleaner code in the future
Problems and Solutions.	Learning a new concept in code is always difficult, however I have managed to create the Page() class which will be inherited by all the subclasses in my application. This is the page class: `class Page():` `def __init__(self):` `self.app = customtkinter.CTk()  # Changed from CTk("Ignatius App") as CTk does not take title here.` `self.frame = customtkinter.CTkFrame(self.app)` `self.app.geometry('400x400')` `customtkinter.set_appearance_mode('light')  # Set appearance mode globally` `customtkinter.set_default_color_theme('green')  # Set theme globally` `self.frame.pack(pady=20, padx=60, fill='both', expand=True)` `for i in range(11):` `self.frame.grid_columnconfigure(i, weight=1,)` `for i in range(12):` `self.frame.grid_rowconfigure(i, weight=1,)`

	It was hard at first to utilise inheritance effectively, and not cause issues. I had to remove all the previous repeated code which was a hard endeavour but it the implementation made the code look mch cleaner.

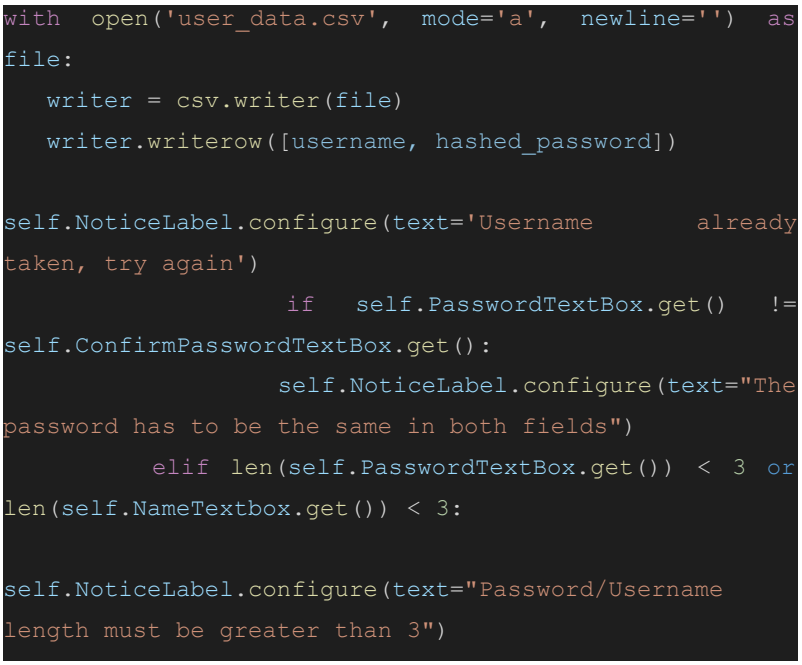
Date:	May 14th To may 17th
Week:	Week 7
Objectives:	The application requires a large amount of images for the test prompts. To remove the mindless data entry component of adding this I have used ChatGPT to create an algorithm that goes onto the HFM website and gets all the images. Once I have gotten these images, I got ChatGPT to sort them into categories and remove images that were not required for the test. Then, I placed the data into a CSV file, with the help of CHATGPT of course. My objective now is to create a test based off the data from the csv file using pandas.
Problems and Solutions.	It was very difficult to understand the way pandas works, and how it loops through CSV files to find things against criteria. I have decided to do this process procedurally at the start of the app.py file. This makes it easier to deal with, when compared to object oriented code. Currently the csv headers are: Name,Category,Popularity,Season,Image Location The category header is so the test is harder, and answers can be in the same category of the prompt. The new algorithm uses a {} list to find four unique answers within the same category. The correct answer is then selected from the previous four selected items and thus the prompt is selected too. The image location is then used to provide an image into the UI, so the test can be completed. `df = pd.read_csv('FruitsAndVegetables.csv', encoding='utf-8-sig') # Reading the CSV file into a DataFrame using pandas, i am unsure of how this works fully.` `fruits_and_vegetables = df[['Name', 'Category', 'Image`

	``Location']].dropna().values.tolist() # Extract the relevant columns (Name, Category, Image Location) and drop any rows with missing values. CHATGPT helped with this.(a bit cheeky) categoryToItems = {} # Initialize an empty dictionary to group items by category, dictionaries are good because i groups items in pairs, which is essential for this usecase for name, category, imageLocation in fruits_and_vegetables: # Loop through the list of fruits and vegetables if category not in categoryToItems: # If the category is not already a key in the dictionary, add it with an empty list as its value categoryToItems[category] = [] categoryToItems[category].append((name, imageLocation)) # Append the current item's name and image location to the list for its category ``` I added the Bcypt library to hash passwords, which is useful because it makes the application far more secure.

Date:	Saturday May 18
Week:	Week 7.
Objectives:	I need to make it possible to traverse through the application, so users can login, and then take their test. I need an accurate hashed password checker too, so users can login and their passwords are hashed.
Problems and Solutions.	I created a login checker that checks users' inputs using the Bcrypt password checker algorithm. It can decode hashes and then check the password. Created a showpage() function in the masterclass of Page, this can be used throughout my application to show any page I wish. ```def DestroyCurrentPage(self): for widget in self.app.winfo_children(): widget.destroy() #the For loop loops through every widjet in the master and subsiquently destroys```

	it. def ShowPage(self, page_class): #this destroys eveytihng and shows a different master/page. self.DestroyCurrentPage() page_class(master=self.app) #here it is changing the page. This is a snippet of the code. So, the for loops goes through each widget and destroys it appropriately. The page class parameter allows the change to any page I wish, e.g. it could be changed to Showpage(LoginPage) and it would show the loginpage appropriately on command.

Date:	Monday 27th May
Week:	Week 8
Objectives:	Currently my way of storing user data is outdated because registration is created through the generation of new files for each user, my objective is thus to change it to a csv file, making it far easier to manage. I might also create a registration validator so users cannot write really long strings, ect. So I don't run into logic errors down the line. Created a homepage two days ago, which currently does nothing except show a popup of score. The score system doesn't work however I will update that later. The homepage will show peoples average score and direct them to the test once completed.
Problems and Solutions.	I still find looping through CSV files quite challenging, however I eventually got it to work on a condition. Because csv files start at -2 it was frustrating as I didn't initially know this. For the csv file to get written into i needed to create a test, test, user. But now it works fine.

	The password and username validation was fine, no issues encountered.
Code:	

Date:	Monday 3 June
Week:	Week 9
Objectives:	Haven't been able to work on the project as of late, making small minute changes and bug fixing recently due to the nature of assessment tasks. However, today I wanted to create an accurate scoring system that would tell the user the score at the end of the test. I might change the score to a global variable as it needs to be used throughout the program, and thus makes it easier if it is global. Do not know though. I need the score to be accurate so later down the line I can make an average score system so users know where they are at in terms of performance. Maybe I will add a manager system too, where managers can see everyone's average. That will have to come later though.
Problems and Solutions.	The issues surrounding score initially, is because the question wouldn't update when you got an answer wrong,

	instead it would update when you got the answer right eventually. So it would give a perfect 20/20 score no matter what. I changed this making it a global variable and created a success page which would inform users on their score. However the issue still remains. The problem is, that the algorithm updates when the score is 20, not when the total question count is 20. I will change this later.
Snippet of code:	`lass SuccessPage(HomePage):` `  def __init__(self, master=None):` `  super().__init__(master)` `  global score` `  self.FVTestButton.place_forget()` `  self.FVTestButton = customtkinter.CTkButton(master=self.MainFrame, #take the test button` `width=500,` `height=50,` `text='Okay',` `font=('inter', 20),` `command=lambda: self.show_page(QuizzPage))` `  self.showscore = customtkinter.CTkLabel(master=self.MainFrame,` `text=(f'Your score is:'),` `font=('inter', 20))` `  self.scorenumber = customtkinter.CTkLabel(master=self.MainFrame,` `text=(f'{score}/20'),` `font=('inter', 100))` `  self.showscore.pack(padx = 25, pady = 30)` `  self.FVTestButton.configure(text='Okay')`

	`self.scorenumber.pack(padx=25, pady=15)` There is also a global score variable at the start, and a quick if statement that is IF score ==20: Show quiz page.

Date:	Thursday 13th of June
Week:	Week 10
Objectives:	The objective was to enhance the user data management by looping through the CSV file to record indexes and usernames for data to be stored later, and to remove the use of global variables. I want to record the indexes to averages and scores can be recorded in the csv file later.
Problems and Solutions.	Implemented a loop to record indexes and usernames for future data storage. No problems so far. I added two arrays at the start of the program called TempDataStr and TempDataInt. These will record the username and index of the csv file. There is also an algorithm that appends to this.

Date:	Sunday 16th of June
Week:	Week 10
Objectives:	Now that the scoring system works, I previously changed it to record to a txt file instead. Using TempDataInt to do so. Then using csv, I recorded the average. This average will be displayed to users. I changed it to a txt file because the scores did not need to be associated with the user as much, moreso just the average. My objective is to create an average score system and make necessary changes to the CSV file to remove errors and improve data handling.
Problems and Solutions.	It was difficult to change, use the data from the txt file and

	generate an average, but with a bit of help from ChatGPT I sorted it out pretty quickly.
Code:	``` def calculate_average(self, line_number): with open('scores.txt', 'r') as file: lines = file.readlines() if line_number - 1 < len(lines): line = lines[line_number - 1].strip() values = [int(value) for value in line.split(',') if value] average = sum(values) / len(values) if values else 0 return int(average) return None ``` No problems with implementing the new UI elements to the homepage, all went relatively smooth. I also changed the categorical sorting system for the csv, adding a misc category to prevent this error where not enough items fit within a specific category.

Date:	Tuesday 18th of June
Week:	Week 11
Objectives:	I want to create a manager mode, as the test part of the app is close to completion, and since there is nothing else to do I may as well add this. The manager will be able to see users' average and latest scores. They will only be able to see users who are falling behind though, because monitoring everyone would be a painstaking process.
Problems and Solutions.	Creating the manager page was easy as well as the checking system for managers. I currently only have one manager registered and I might add a way for managers to add more managers. I want to add a way for managers to notify users however the time constraints may forbid this.
Code:	**Check the commit on this date as the code is pretty large so it is difficult to fit it in this box.**

Date:	Thursday 26th June
Week:	?
Objectives:	With the due date being tomorrow, I am cleaning up the code in preparation for assessment. I have decided to make the manager page and clean up several aspects of my code through the file management.
Problems and Solutions.	Overall the development of this project has been good, with the last objectives being relatively easy, a readme file has been set up with instructions on how to use the app.

Test Table:

Test ID	Category	Test case description	Input to provide	Expected output	Actual output	Pass/Fail	Action taken
1	Path coverage	Verify attempts counter increments correctly	Provide incorrect passwords repeatedly	Counter increments until reaching max attempts, then lockout	Counter increments correctly and locks out after max attempts	Pass	N/A
2	Boundary Value	Check behaviour on last password attempt	Provide correct password after several wrong	Login success after correct attempt	Login success after correct attempt	Pass	N/A
3	Faulty Data	Input non-alphabetic characters as username	Input: 'user!@#'	System rejects invalid username	System accepts invalid username	Fail	Updated input validation to reject non-alphabetic characters.
4	Path Coverage	Ensures that the quiz accurately detects a win when the user provides the	The correct answers in the proper order.	The quiz should continue when the user correctly gets	The quiz continues and indicates a win through	Pass	N/A

		correct answers in sequence without exceeding the maximum number of allowed attempts.		the answer in sequence progressing to success.	going to the next question, then success page.		
5	Abnormal Data	Enter already registered username	Provide existing username during registration	System notifies username already taken	System allows duplicate usernames	Fail	Fixed the system to check for existing usernames and notify the user accordingly
6	Data Integrity	Ensure data is saved correctly in CSV	Register multiple users	Users' data should be saved and retrievable	Users' data is saved and retrievable	Pass	N/A
7	Performance	Check system's performance under load	Simulate multiple concurrent logins	System should handle load without crashing	System handles load without crashing	Pass	N/A
8	Security	Verify password hashing	Register user with a password	Password should be hashed in CSV	Password is hashed in CSV	Pass	N/A
9	Usability	Check user interface response	Navigate through the application	UI should be responsive and intuitive	UI is responsive and intuitive	Pass	N/A
10	Functionality	Ensure score calculation accuracy	Complete a quiz, counting the score	Score should be calculated and displayed accurately	Score is calculated and displayed accurately	Pass	N/A

GitHub Repository:

The information regarding the readme is within this repo:

https://github.com/IggyMeagher/major-project.git

Development Reflection:

The original aim of this project was to create a basic training application for Harris Farm Markets employees, or other retail stores, to enhance their knowledge of fruits and vegetables. The main objectives were to create a system where users could take quizzes, view their scores and averages, and enable managers to monitor employee performance. The application intended to streamline the training process, reduce the time and cost associated with it, and alleviate stress for new hires. Several aspects of this project were successful, with the main quiz logic achieving its outlined purpose in the problem definition, effectively giving users 20 questions on different fruits and vegetables. Furthermore, I have developed an algorithm to these questions, in which answers are based on categories, to make the test more difficult. Users can now take quizzes, get immediate feedback on their performance, and have their scores updated to a database. Additionally, the application calculates and displays average scores, providing users with insight into their progress. The login and registration functionality, including secure password hashing using Bcrypt, was also successfully integrated. A user centric application was also created, using best practice coding to achieve this, with the help of customtkinter, and other API functionalities.

The project faced many challenges during its development, and indeed, there are multiple places in which the project can be improved. Specifically, the data only contains about 160 possible fruits and vegetables, which is underneath the scope outlined in the problem definition. This is due to the issues surrounding the sorting of data and faulty data that comes with scraping from websites. During the original planning of the project, the objective was to create a more intense algorithm that gave users tests based on their previous answers, however, as the application development progressed it was found to not be feasible under time constraints. Furthermore, a forgotten password system would have been beneficial to be implemented, but again, time constraints prohibited this. Additionally, more manager controls could have been implemented, where users could be notified on their progress, however, instead the project opted for a in person feedback approach, where managers can see users who are falling behind and thus give feedback. There were a plethora of problems faced during the applications development such as: the learning of the panda's library was difficult, with the methods that come with it being hard to learn. Specifically, I found looping through csv is particularly difficult due to the nature of the different columns and rows. This was the hardest when an average had to be fetched, or an admin had to be verified, in which a specific user's data had to be retrieved. This was hard due to the alien nature of the api, as well as the data structure that was the csv. Overall, during the development, multiple problems were introduced throughout the SDLC process which is typical in regards to the nature surrounding the creation of software.

Overall, these challenges have developed my understanding of object oriented code, different libraries and python. Helping me understand the importance of clean, well structured code that does not repeat itself. The introducement of version control has allowed insight into the programming world after school, and the devices used within it to maintain versions. So, in conclusion, my development as a software engineer has progressed significantly due to the learning process of this assessment task increasing my understanding of code itself, and how it works in relation to data structures and version control.