

# OCR GCE A COMPUTER SCIENCE PROJECT H446-03

---

Name: Naveed Ali Rafeeq  
Candidate Number: 1904  
The Solihull Sixth Form College: 20780  
Title of Project: AS Physics and Mathematics Quiz

# H446-03 – PROJECT CONTENTS

## TABLE OF CONTENTS

A. Analysis.....	3
B. Design .....	20
C. Developing the coded solution (“The development story”)	55
D. Evaluation.....	143
Project Appendixes.....	184

## A. ANALYSIS

### Identifying Stakeholders:

My project is a multiple-choice quiz that covers the two AS subjects consisting of Maths and Physics. It is a quiz where students will have a username and a password, this is used to personalise the quiz to tailor their skills and weaknesses. Each subject will be divided into topics and the quiz will have questions that are included in each of these topics, therefore if the user is not doing so well in one topic then the quiz will give them more questions from that topic. E.g. If the student does a physics quiz consisting of 20 questions from 4 topics, and does well in Particles yet not so good in Mechanics. Then, next time he takes the quiz he will be given more questions from the Mechanics topic and less from the Particles. Furthermore, a special login will be given to teachers so that can see how well the students in their class are doing and will also have a visual leader board to look at, so they can see which students are doing well and which aren't.

Due to the difficulty of the quiz it appeals to first year college students since the quiz is designed to help them practice and improve their Physics and Maths skills. It is also tailored for college teachers to use since they can approve the quiz to their whole class and can therefore keep track of their progress through the leader boards system. The reason why the audience will use my system is due to the fact that unlike other quizzes not only can the teacher edit the questions inputted but the quiz also changes without informing the student. This means that the students will always be getting challenged since they are being questioned on topics they are not good at.

One of my stakeholders is a year 12 student called Moaz Masood, he is currently studying Maths, Physics and Computer Science. The other day he was complaining about how he could not find a system that can test his maths ability that you can do in your own time whilst being marked and overlooked by a teacher as well. So, he asked me if I could help him and other year 12's by making this software for their whole class to use. My quiz will meet his needs because it will test all of his maths abilities and then the teacher can help him out because they can see where he's going wrong.

Another stakeholder is a Physics teacher at my college as well as a private Maths tutor, he was explaining to our class that he uses a website called Kahoot to test his students on their Physics knowledge. He said it was good however it could have been better, therefore I interviewed him at the start of the year to find out the about the system and how it could be more bespoke for a teacher needs. My quiz will benefit him due to the fact that he will be able to choose what

Candidate Name: Naveed Ali Rafeeq  
questions to give to his students, but they don't have to be in the lesson live but instead it will be integrated within the database of questions.

Candidate Number: 1904

### **Thinking Abstractly:**

My quiz will contain a large amount of abstraction in order to fit the stake holders needs and makes it simplified for them. In the quiz, All the calculations have been stripped from the project, this is due to the fact that it is not needed within the project since all that is needed is the answer of the question. However, within real life the answer is not the only thing that is needed due to the fact that with most mathematical based questions working out is needed. Thus, in real life problem solving tests you need to show your calculations in order to obtain most of the marks but the abstraction in the quiz has removed the marking of that and has replaced it with a scoring system instead, which brings me on to my next abstraction.

The scoring system within my quiz is an abstraction since there are a variety of questions all at a different difficulty level. In real life these questions are all given a range of marks for answering them. Never the less, my quiz only gives out one point per question answered correctly no matter what the difficulty. This has allowed me code my quiz with much more efficiently since they don't each have their own bespoke mark. Also, it makes the quiz easier and simpler for the stakeholders since they track how many questions they got wrong proportional to how many marks they lost.

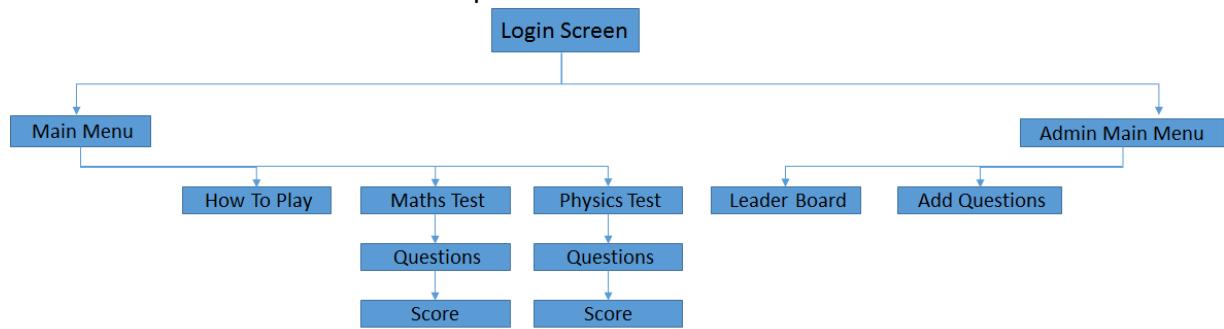
### **Thinking Ahead:**

There will be a variety of Inputs and Outputs within my quiz. I thought about these ahead of coding the quiz due to the fact that I need to understand the problems and implications of the inputs and outputs and how to fix them.

There will be many parts of my quiz that I will re-use, from the graphics to subroutines of the code. This will help me significantly since it is easier to re-use parts of the project instead of writing it all out again. Thus, since that part of the quiz has been tested as well so I will not need to test the code in order to check if it's working. Which will save me a huge amount of time whilst coding my project.

The controls to my game are very simple since it is not a game but just a quiz. Therefore, all buttons will be displayed on screen and the mouse will interact with these. Plus, the keyboard will only be needed when typing in textboxes e.g. Logging in to the quiz.

### **Thinking Procedurally:**



Within my program you can predict the order of it except for at two points. The first is at the login screen because according to your login it will take you to the user main menu or the admin main menu. The second is at the main menus (User and Admin), after these the procedure of the program is just a series of steps that the user undergoes.

Since the quiz I'm creating is a Graphical User Interface (GUI) this means that I will need to use a lot of components within the window forms. Some main components of the program include, buttons to choose an option or to confirm your choice on a question, text boxes to write your user credentials when logging in, labels to inform the user on what some components do and to display the question and the different answers.

Many sub procedures are needed within my quiz, one of the main sub procedures are for the questions. Most questions will be pulled from databases therefore a sub procedure is needed to use SQL in order to obtain a random question from it. Another main sub procedure needed is the leader board since the code will need to run a sort algorithm to order the scores of the users.

### **Thinking Logically:**

One of my main decisions within my program is to verify whether the login details are correct. If they are matching any of the usernames and the linked password connected to it then the application will allow the user to continue onto the main menu, if not then the code will display an error message and tell the user that they have inputted the wrong details. It will then loop it back to where the user can input their details in again.

Another example of a logical decision within my quiz is the main menu, since there are a variety of options to choose from this means that the user can choose what the program does next. So, if they choose maths the program will load the maths section of the quiz and that subroutine will run.

### **Thinking Concurrently:**

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

An example of a concurrent process within my program is when the user is logging into the quiz. It is used here due to the fact that it needs to use an AND function in order to check whether the username and password is in the database. It has to be done concurrently because both the username and the password need to be found at the same time.

Another example of concurrent operation within my quiz is when the user answers a question. When the user clicks the button to confirm his answer a message box pops up informing the user whether the answer was write or wrong, whilst this is happening the program is loading another question from the question list within the database for the user to attempt in the next form. The benefit of this is that the user will not need to be waiting between each question whilst the new question loads instead he/she can read the message box informing them of the correct answer. So, when the user closes the message box the new question has already loaded.

### **Hardware & Software:**

#### **Software needed are:**

Visual Basic: I am going to be creating my quiz on visual basic since its Graphical User Face (GUI) is very easy to use compared to other languages, for example on Visual Basic you can drag drop and resize all the tools used in the GUI however in Python for the GUI you have to manually input all the tool dimensions which makes it more difficult to use. Also, since I have been using Visual Basic for the past year and have learnt all the GUI skills necessary for this project on it its best that I create the quiz using Visual Basic.

Microsoft Access: I need a database to connect to in order to store all my data related to the quiz lie the logins as well as their scores and questions. I am using Microsoft Access because it is one of the most used database software as well as it being the one I am most comfortable with.

Operating System: Windows 7 +, Visual Basic only runs on Windows operating systems therefore it is needed.

#### **Hardware needed are:**

Mouse: A mouse is needed in order to perform all the actions within the quiz since you only need to click on things.

Keyboard: Is needed to type in your user name and password at the start login screen.

These are the Computer requirements needed to run Visual Studios 2017, which Visual Basic is run on. This is displayed on their website

**Hardware**

- 1.8 GHz or faster processor. Dual-core or better recommended
- 2 GB of RAM; 4 GB of RAM recommended (2.5 GB minimum if running on a virtual machine)
- Hard disk space: up to 130 GB of available space, depending on features installed; typical installations require 20-50 GB of free space.
- Hard disk speed: to improve performance, install Windows and Visual Studio on a solid state drive (SSD).
- Video card that supports a minimum display resolution of 720p (1280 by 720); Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher.

<https://www.visualstudio.com/en-us/productinfo/vs2017-system-requirements-vs>

Even smaller Hardware requirements are needed to run Microsoft Access:

- 1 GHz processor
- 256 MB of RAM or higher
- 3 GB available disk space
- 1024 x 7698 Display or higher

These are all found on Microsoft's website

[https://technet.microsoft.com/en-us/library/ee624351.aspx\(v=office.14\)#section3](https://technet.microsoft.com/en-us/library/ee624351.aspx(v=office.14)#section3)

**Research:**

I conducted an interview as part of my research with one of my stakeholders Brendan, I asked the stakeholder what problems there are with a current quiz based in these subjects.

Q: Have you used a multiple-choice quiz before to allow students to improve on their Physics and Maths skills?

A: Yes, I used this quiz creating software called Kahoot It was good since it had custom questions and not ones found in past papers or in books with answers at the back, this allowed the students to have to attempt the question instead of searching for the answers. It also had an accessible interface it was clear what to do and the buttons where easily visible.

Q: Was there anything that you didn't like about Kahoot

A: Yes, unfortunately the quiz would take a long time to create since you had to manually put 20 questions into the quiz. Also, everyone had to be present and have an internet device with

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

them at the time. This makes playing the quiz awkward, because not only do you need constant internet connection but also everyone to turn up to the lesson, which is an uncertainty.

Q: What would you change about ... If you could?

A: The main issue is the fact that the quiz must be live, if I could set it out to everyone and they did it at home then I could check the leader boards who did well and who didn't then the quiz would be much better. And since Kahoot is used for any type of quiz it would be more useful if there was a quiz out there that was created for Maths and/or Physics where I could possibly edit and add questions to the system.

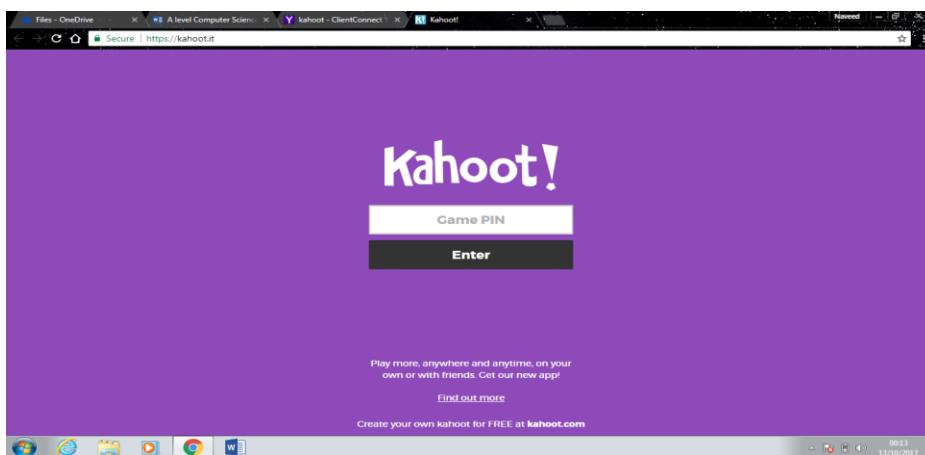
Q: Does a colourful and visual quiz make it better?

A: Not really no, it is mainly the quiz that matters and all the colours and graphics are just extras.

Q: How many questions do you think is appropriate for each time attempting the quiz?

A: I would say around 20 is great, since it's not too long but long enough to see student's weak topics.

<https://kahoot.it>

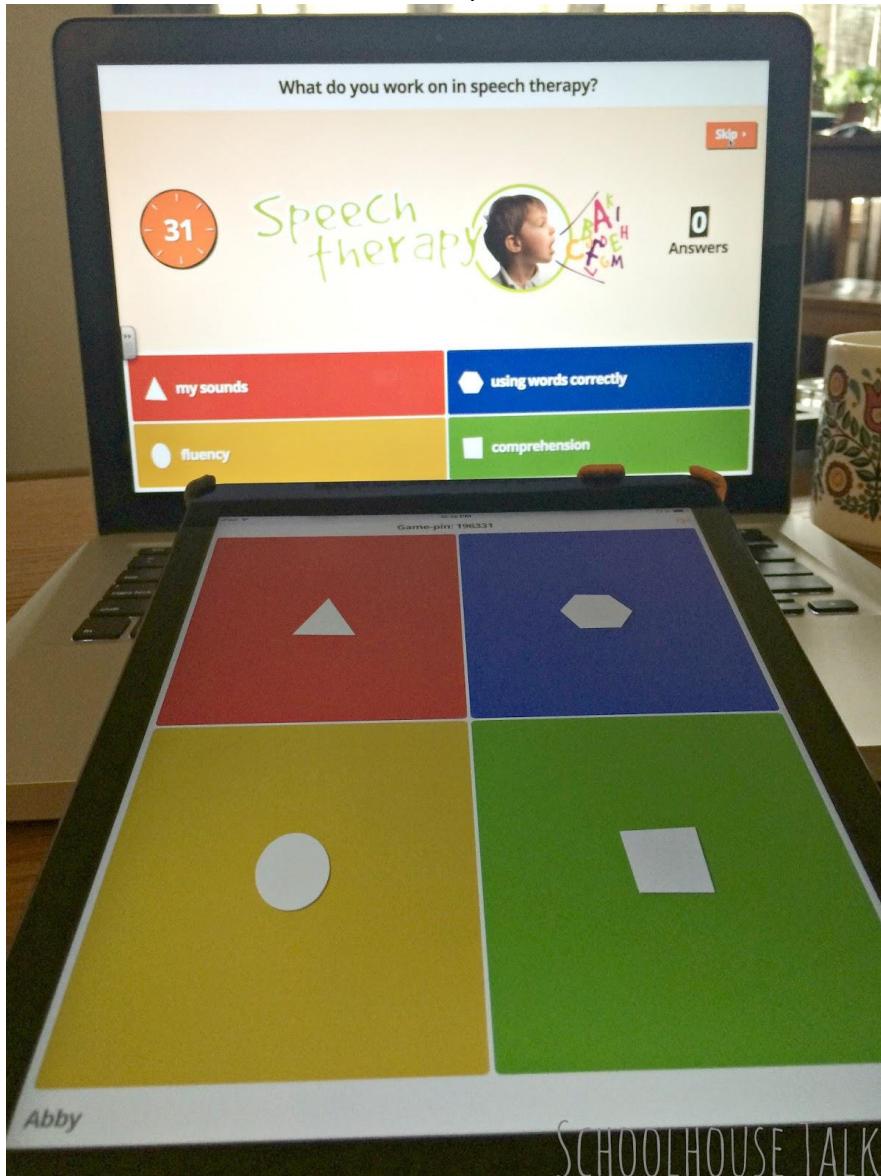


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

The screenshot shows a Kahoot! game lobby. At the top left, there's a call to action to "Join at kahoot.it" with a Game PIN of "9928641". On the right, there's a "Full Screen" button. Below this, the lobby interface shows "145 Players" and a large "Kahoot!" logo. A "Start" button is visible in the top right corner. The player list is displayed in three columns:

Player Name	Player Name	Player Name
LoRCaN't	kiM-JOnG Uno	BEn doVeR
LoRcaN'T	EILEEN DOVer	EILEen DOVeR
ReALpErsOn.Mp4	Not A BOt	LoRcaN't
lorCaN't	sPORTaCuS	SeTUP.ExE
SPORTacus	LOKESH;)	ur sEArcH hISTo
not iN uD ClAss	adMiN1	z1cooV



<http://slp123.blogspot.co.uk/2015/08/kahoot.html>

Kahoot is a live quiz that many people can join and participate in, the person who created the quiz will have written many questions that all the participants will answer within the time limit, the quicker they answer the more points they obtain. All these points will accumulate as well as the fact that you get extra points for getting multiple answers right in a row. At the end it shows everyone's scores and position in a leader board and then, if the creator of the quiz decides, they can save the leader board into a spreadsheet. So that it can be viewed at a later time.

This is a very good website that allows the users to have a lot of fun as well as testing their abilities within the subject. There is a short time period as well as the leader board and your position being shown after each question, this causes immense competition within all of the

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

users which affectively creates test like conditions that will prepare the user for the real AS exams at the end of the year. Another thing that I liked about Kahoot is the fact that all of the questions are bespoke, this means that the teacher can choose questions that they know the class is averagely bad at and need practice on.

But even though as good as it is the main problem of the quiz is the fact that all of the users participating have to be there in the room while it occurs, they need to have visuals on the main board where the questions are displayed. This is very annoying since there is a very small chance that there will be 100% attendance every time the quiz occurs, this means that certain students will not be tested and therefore they miss out. Another problem is the fact that certain people may not be comfortable with their position on the leader board because they can be embarrassed by it, thus the students with better scores could possibly bully those with lower scores, therefore it may be better if only the teacher can view this leader board and see who is doing well and who is not.

My quiz is similar to Kahoot due to the fact that it creates test conditions since it has a certain amount of questions of different topics that the student will have to answer, just like the AS examinations at the end of the year especially for Physics because there are whole sections of multiple questions just like mine. However, my questions will not have a timer since the questions are harder and need a long time to solve so a timer will restrict this. Another similarity is that my questions are all stored within a database therefore I can use SQL within the quiz so that the admin account can edit and add questions to this database and therefore, creating bespoke questions. Furthermore, all of the students' scores will be saved to a database and therefore sorted using a sort algorithm when loaded into the leader boards. This way the admin can keep track on each of his students' progress in each topic. The advantage my quiz has over Kahoot is simply the fact that it can be done in the user's own time whenever they want, which is what Brendan said within the interview. All in all, even though Kahoot has amazing visuals and is a great website it is however too simple for the requirements that is needed for students at AS level and therefore it is not as useful to them as my quiz will be.

I also performed another interview with my other stakeholder Moaz, he currently uses this website called Isaac Physics set to him by his teacher to revise for Physics, the interview went as following.

Me: So, what is Isaac Physics?

Moaz: It is a website made for Physics students in order to test their A Level Physics knowledge using hard complex problems that you have to answer in order to complete the test. Therefore, you are given multiple attempts on the same question.

Me: Alright what do you think stands out for you when using this website?

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

Moaz: The fact that your work can be assessed by the teacher that gives it to you as well as they choose what topic the questions are from, this allows the teacher to give you topics that you are bad at and thus allowing you to work on them.

Me: That's great, is there anything you would change about it?

Moaz: Yes, I don't like the whole way the quiz is set out, it is more like answering questions from the back of a textbook but without the answers. The questions are so hard and yet they do not have the answers, the amount of time I have spent just on single question is ridiculous. They should change it so that it is more like a test environment where you get marked and then it shows you where you went wrong, also once it is completed you cannot do it again because it is not like a test therefore I have either completed it very quickly with ease or I have been stuck on the same topic for hours on end, there is no middle.

Me: Alright thanks for that Moaz.

Moaz: No problem.

I also researched the Isaac Physics website myself, here's what I found.

<https://isaacphysics.org>

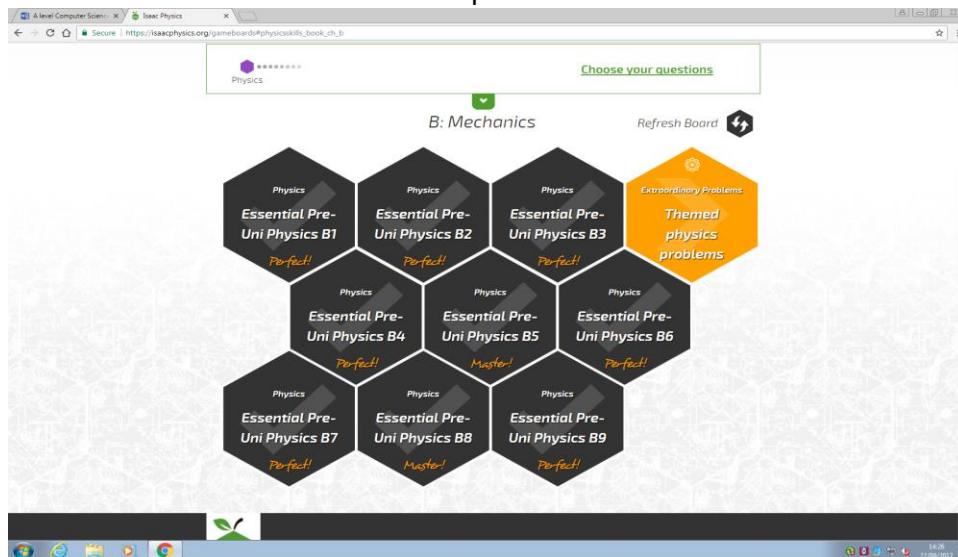
The screenshot shows the 'My Assignments' section of the Isaac Physics website. It displays four assignments:

- D: Waves**: Progress 93% (purple bar). Due 10/07/2017 by B. Foster. Contains 3 questions: Q1, Q2, Q3. View Assignment.
- C: Electric Circuits**: Progress 94% (purple bar). Due 10/07/2017 by B. Foster. Contains 3 questions: Q1, Q2, Q3. View Assignment.
- B: Mechanics**: Progress 97% (purple bar). Due 10/07/2017 by B. Foster. Contains 3 questions: Q1, Q2, Q3. View Assignment.
- G: Gases and Thermal Physics**: Progress 89% (purple bar). Due 10/07/2017 by B. Foster. Contains 3 questions: Q1, Q2, Q3. View Assignment.

The page includes a navigation menu, search bar, and footer links for About Us, Why Physics?, Extraordinary Problems, and system status (144.1, 22.09GB (0%)).

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



The screenshot shows a question titled 'BS.1'. The question text: 'The strength of Earth's gravity at ground level =  $9.8 \text{ N kg}^{-1}$ . 1 tonne =  $1000 \text{ kg}$ '. Below the question: 'Numeric answers should contain units. Where forces are asked for, ensure that the direction is in the answer (e.g. up/down). Assume that the mass is evenly distributed in the rulers, poles, planks, bridge spans mentioned in the questions.' In the 'Part A' section, there is an 'Answer Now' button. The user has entered '4.5' in the 'Value' field and 'N' in the 'Units' field. Feedback: 'Correct!' and 'Well done!'.

This site consisted of users and an admin, all the users when signing up will have to write the admin code that will link their accounts. Then, the admin sets out assignments to the student users and they have to complete these set of questions based on each topic. Each user will have to then complete each topic to a minimum of a mastery level, which is usually around 80 - 90%, in order to complete it. Then when you complete all the sections in one topic it notifies you that you have mastered the whole topic, once you have completed every single question within the topic it notifies you that you have perfected the topic and places it into the completed assignments.

The site is good due to the fact that unlike Kahoot, the users don't all have to be present in order to participate. Thus, it has no time limit so it can be done whenever the user desires which makes it good for a homework task. Another good point about Isaac Physics is the fact that it has a friendly user interface that is not only visually appealing but it also gives the user

tips whenever they get the answer wrong, which allows the students to improve their answer as well as obtain a better knowledge of physics as well in order to help them with their main aim which is their end of year exams.

However, there are some parts of Isaac Physics that I don't like. The main problem is the fact that it allows the user to attempt the same answer more than once. This is not that useful for the teacher since the teacher would like to know what the weak points of each of their students are. So, if the student keeps on attempting the same question loads of times and then finally gets it right then by guessing then it negates the purpose of the question. I know this due to the fact that I did this a few times myself when completing the questions on Isaac Physics.

My quiz is similar to Isaac Physics in a few ways, one is the fact that it has a one to many relationships between the teacher and the students. This allows the storage within the database to be easily grouped and organised into subjects and topics and linked to the students. Another similarity is that the questions are split up into topics just like with Isaac Physics, this is helpful due to the fact that the teacher will have all the topics linked to their account and then they link those to topics to all of their students to complete. A disadvantage is that with Isaac Physics is that the quiz is only completed when all of the questions are answered correctly. However, with my quiz it will be completed after 20 questions and afterwards the score will be displayed accordingly. This is advantages since it allows the student as well as the teacher to be able to see how well the student actually is performing.

### **Features Of The Proposed Solution:**

- Before the quiz main menu starts there is a login screen that all users must login to in order to access the quiz. If they do not have an account yet then they will have to create an account and also find out their teachers link code in order for their teacher to view the students' progress.
- The quiz will consist of mainly 2 parts, the Maths quiz and the Physics quiz, each will be unique in their questions and will both have a different number of topics. How well you're doing in one will not affect the other quiz therefore making them completely separate.
- The main menu will come after the login screen and this will display the title in bold and have 5 buttons for the user to click on.
  - How To Play: This will open a new window that will inform the user on how to play the quiz as well as how it works. Plus, all the extra features that are included within the teachers account.
  - Start Maths Quiz: This will start the Maths part of the quiz where then the user will then choose whether they want to do a quiz on Core 1 or Core 2. Then they

will answer 20 questions from different topics within either C1 or C2 and then afterwards it will send them back to the main menu.

- Start Physics Quiz: This will start the Physics part of the quiz where the user the user will be tested questions on AS physics topics of the following; Mechanics, Materials, Electricity, Waves, Quantum Phenomena and Particles. After performing those 20 questions it will then also loop bac to the main menu.
- Change User: This button will send the user back to the login screen allowing someone else to login instead.
- Exit: This button not only just exclusive for the main menu but it appears everywhere on the quiz in the same position allowing you to stop the whole application whenever you want.
- Each part of the quiz will have a different background and theme to it. These themes are linked to whichever part of the quiz you are on, for example if you're doing the physics quiz and your performing an Electricity question then the theme of the windows form will be linked.
- If I had more time I will would have added more subjects to the quiz in order to make it appeal to a wider range of stakeholders. This would not just be at AS level as well but from GCSE to A2.
- The teacher's main menu is a different to the students since it has 3 options, these are:
  - Math's Score Board: This will have ranked all the students Maths scores using a sort algorithm as well as displaying what they got in each test. This allows the teacher to see who's doing well and who needs more help.
  - Physics Score Board: Same as the one above but for Physics instead.
  - Customize Questions: This button actually opens a new window that allows the teacher to edit the questions that are stored within the database as well as adding more questions of their choice. However, they cannot change certain questions due to the fact that they rely on Random Number Generators (RNG) within the code to generate the question as well as the answer.

### **Limitations:**

There are a few limitations within my project that are stopping me from creating a better quiz than the one proposed. Ideally, I would like my quiz to be based online so that it would be easier to access as well as the fact that people won't have to download my program. Everyone has internet access so it they could just go to the quiz website and perform the quiz there. However, I cannot to this due to the fact that I would have to make a website to go with it as well as maintain a domain name which has monthly costs. Thus, I do not have the time to create a website as well as the program therefore I cannot make my quiz become online based.

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

Also, I do not have an online cloud to store questions in a database so I have to use an offline database stored within my files.

I would also like to increase the range and the variety of the quiz stakeholders so that it is not only AS students but GCSE as well as A2. But, this is not possible because I simply do not have the time to cater for all these audiences. Thus, I do not currently obtain the knowledge to not only code such a quiz but also to write the A2 questions since I am currently in the 2<sup>nd</sup> year of college myself, therefore I will most likely not know the answers to a few questions.

### **Success Criteria:**

Requirement:	Justification:	Reference:
User/Staff Login before loading up the quiz menu	When the application loads the first thing that appears is the login screen where the users will need to either login using their user me and password or create an account. This will be used in order to save the users score to their previous score in the database. The staff login will be used by teachers in order to see how their students are performing.	Interview
Quiz Menu	This is displayed in order for the user to choose which subject in the quiz he wants as well as the instructions.	Proposed Solution
Scoring System	A point is awarded for each question that you get right, the points are tallied out of a total and then saved to the user so that their scores are saved for later use.	Proposed Solution
Exit Button	Stops the whole quiz and closes it down and stops the code running. Therefore, users can leave whatever time they want.	Proposed Solution
The quiz is navigated with	The reason for using the mouse is	Interview

the mouse	because it will allow the user to click on certain options within the quiz with the most ease. If they use the keyboard then it will be harder to navigate for most people.	
A menu with 5 options	After the login screen, there is 5 options that lead you to a Maths quiz/a Physics quiz/How to play/Change account/Exit. Only these 5 options will be needed to access everything within the quiz.	Proposed Solution
The font size is above 14	To ensure that the writing is visible to the user. So that navigation within the quiz will be easier.	Proposed Solution
There are 20 questions in each quiz	So that the quiz is not too long or short, allowing users not to get too bored quickly.	Interview
There must be more than one topic included within the quiz.	So that the user is attempting questions from a variety of topics.	Proposed Solution
A Message Box appearing after answering each question	To tell the user whether they got it right or wrong, this will allow the	Proposed Solution
Radio Buttons	For the user to only be able to click on one answer at a time and not on multiple.	Proposed Solution
When the user finishes the quiz, a display should appear telling them what score they got out of 20.	This allows the user to know how many questions the user got wrong and right without having to count throughout the quiz themselves.	Proposed Solution
After closing the message box, a button appears to allow the user to go to the next question.	To proceed to the next question. Also, the stakeholder will be able to evaluate the question and understand how the user went wrong.	Proposed Solution

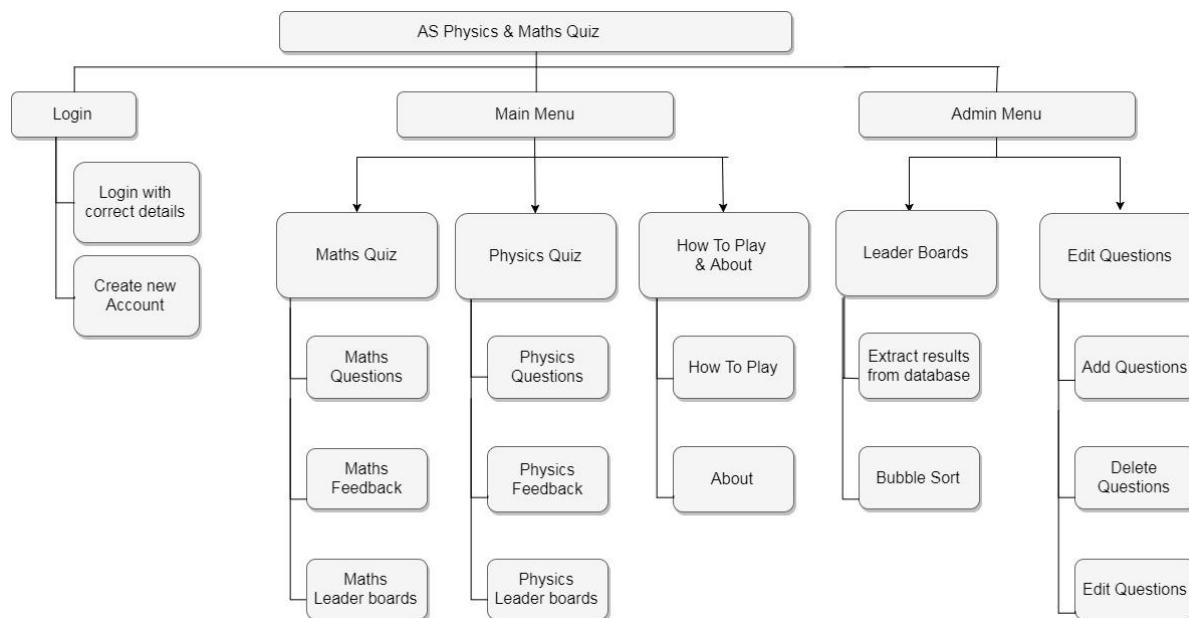
Menu should be easy to use	The Menu will contain the title and 5 buttons therefore the buttons will be large with large fonts, this will allow the user to be able to navigate around the menu with ease and select the option they want.	Proposed Solution
A title should be displayed on the Menu	The title will inform the user what the name of the quiz application is called.	Proposed Solution
Appeal to ages 16+	This is because it is aimed at first year college students so anyone younger shouldn't be playing the quiz since it will be too hard for them therefore they will be likely to fail it.	Proposed Solution
Should display a message at the end of the quiz next to your score.	Different friendly messages will be displayed according to how well they did. This quote will encourage the user to play again by either motivating them or praising them.	Proposed Solution
A grade will be displayed at the end of the quiz.	<p>At least:</p> <p>80% Correct = "A"</p> <p>70% Correct = "B"</p> <p>60% Correct = "C"</p> <p>50% Correct = "D"</p> <p>40% Correct = "E"</p> <p>Anything Less = "U"</p> <p>This is important because then the user will understand how well he did in the quiz. Also, there is no A* grade due to the fact that there is such grade in AS Levels.</p>	Proposed Solution

The teacher's will have an extra 2 options, Maths and Physics leader boards.	Allows the teacher to see the progress of their students in each subject and who is doing the best and who's doing the worst.	Proposed Solution
A sort algorithm to arrange the order of the leader boards.	This will make sure that the leader board is always correct and that it can be done automatically as soon as someone finishes the quiz.	Proposed Solution

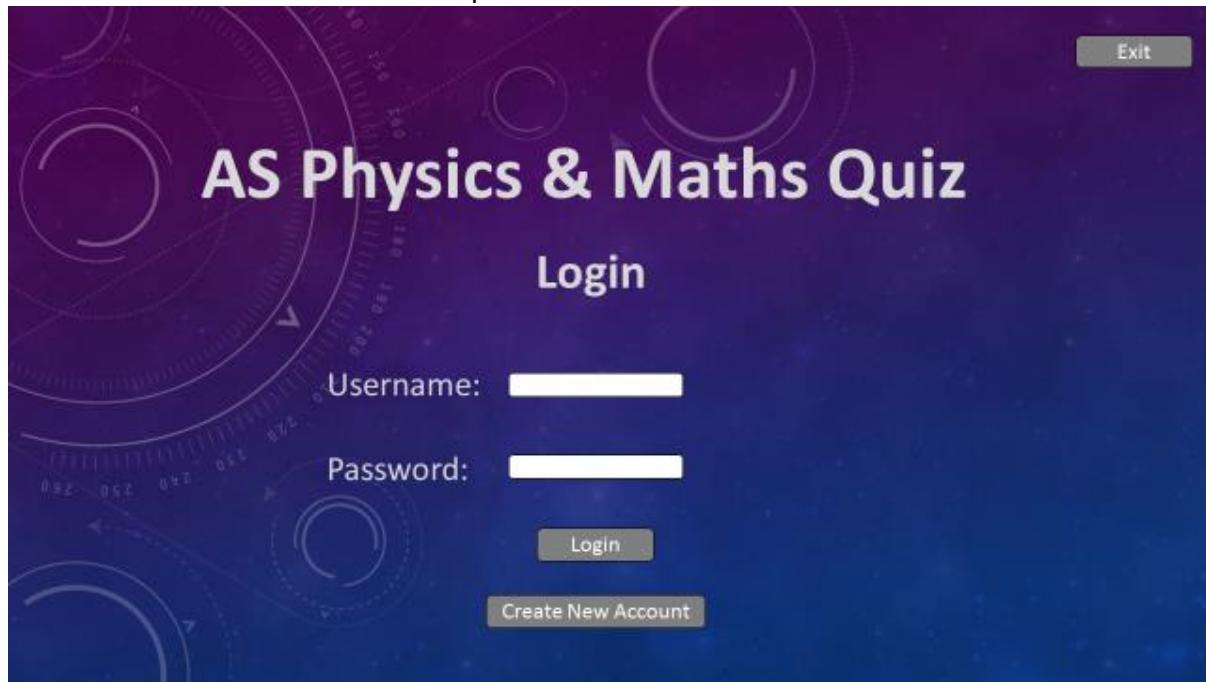
## B. DESIGN

After my interviews and my proposed solution, I broke down my whole quiz and decided to section the whole problem into smaller and less complicated tasks. By doing this I will be able to visualize how exactly I will create my quiz, this method to approaching problems is also known as decomposition. This is important due to the fact that without it I would not be able to know exactly what code I need in each section of the quiz. I will do this by first designing all my windows forms for the quiz, after I have done that I will know what code is needed to run each form. Then, I can design functions for each button within each form and then link all the forms together at the end to create the whole quiz. By doing this I will be able to create each function separately and then be able to just call upon it whenever it is needed within the forms within the quiz.

### Systems Diagram:



### Form Design:

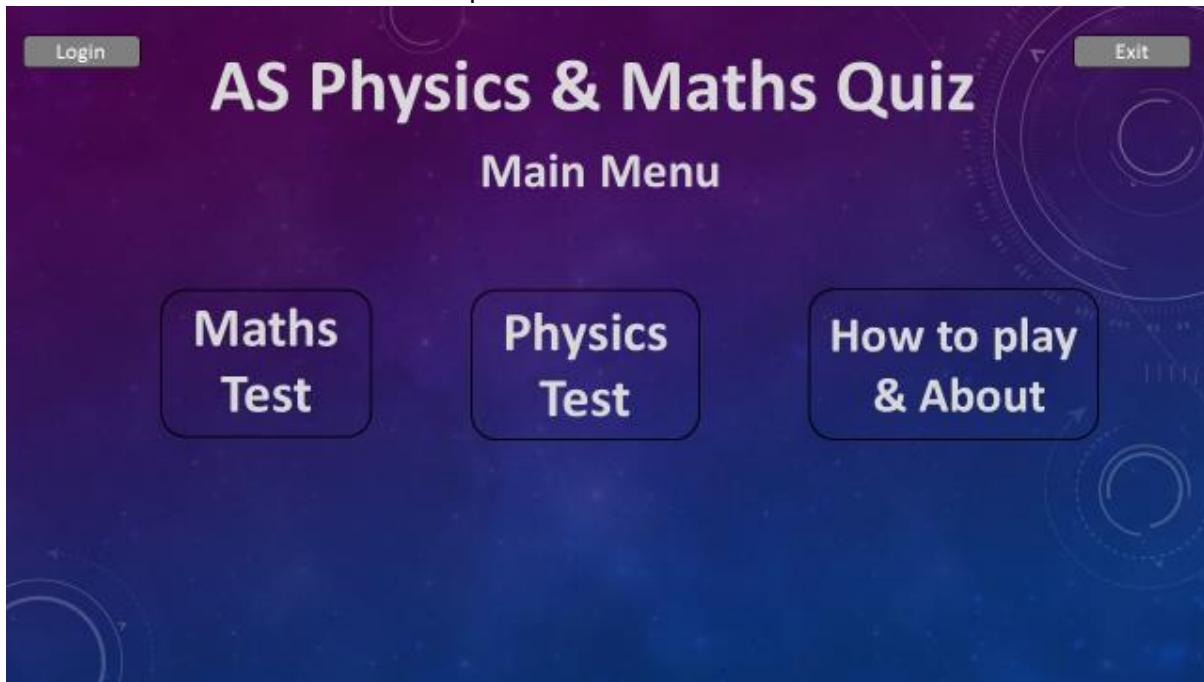


The title 'AS Physics & Maths Quiz' is displayed at the top of the form in large bold font, this is so that it is visually stimulating for the user as well as it being clear what the quiz is called.

The sub title tells the user what form page they are currently on therefore this is the Login screen

The username and password textboxes are labelled so that the users know where to input each of them. The login button takes the user to the next form if the login details match with one in the database. The create new account button allows the user to create a username and password that will be saved to the database, which they can use to login next time.

The exit button stops the whole code from running and closes the running codes. It is quite small and placed at the top right so that people don't accidentally click it, if so it will display a message box to confirm whether they want to exit or not.



The same title is still displayed yet just moved higher to make space for the main menu.

The sub title now displays 'Main Menu' which is the current form page they are on.

The exit button is still placed at the top right yet there is another button on the top left displaying 'Login', this button is displayed in order for the user to return to the login screen, like with the exit button it will ask for a confirmation just in case they accidentally clicked on it.

There are 3 options as part of the main menu form the 'Maths' and 'Physics' tests will both lead to their respective quizzes. The 'How to Play & about' button will inform the user on how the quiz works as well as who made it. All of the buttons will open another form.

The screenshot shows a mobile application interface for a Maths Quiz. At the top left is a "Main Menu" button, and at the top right is an "Exit" button. The main title "Maths Quiz" is displayed prominently in large white font. Below it, the subtitle "Question: 1 / 20" is shown. The question itself is: "What is dy/dx of y = 2x^3 + 8x - 1 ?". Four options are listed below, each preceded by a radio button:

- $2x^2 + 8x$
- $3x^2 + 8x$
- $6x^2 + 8x$
- $6x^2 + 8x - 1$

At the bottom left is an "Answer" button, and at the bottom right is a large "Next" button.

The title has now changed into 'Maths Quiz' which informs the user which quiz they are currently undertaking, this will stay fixed throughout the whole duration of the Maths quiz.

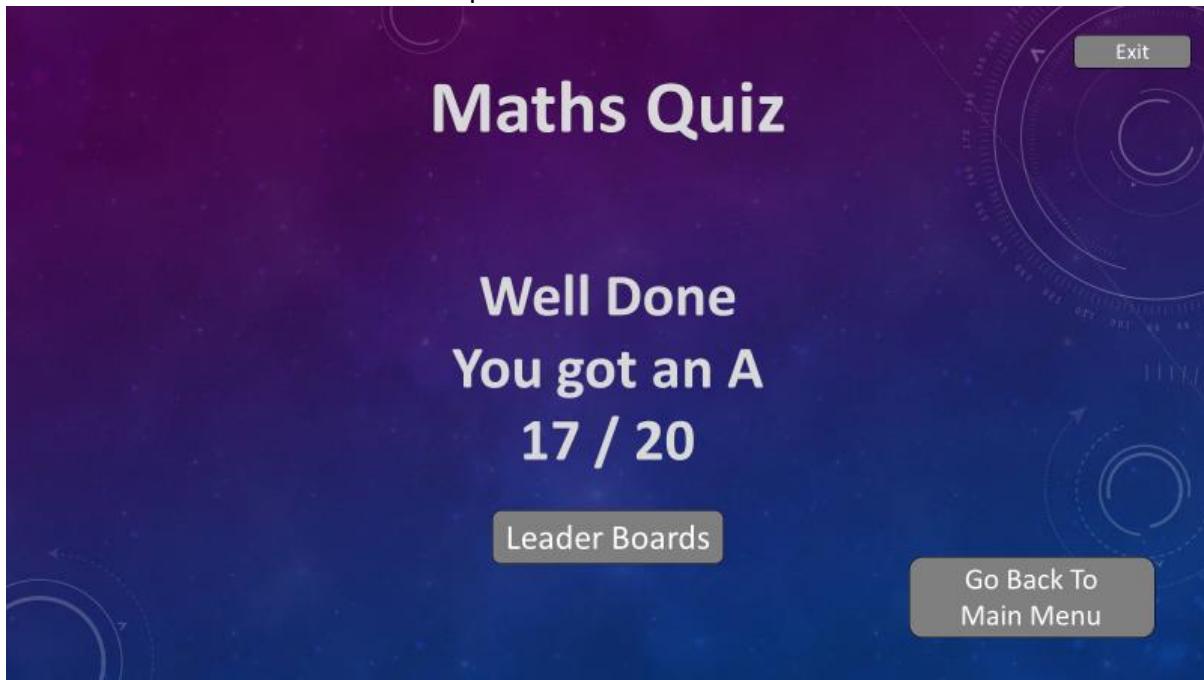
The sub title informs the user which question they are on out of the total questions in the quiz.

A question is displayed below the subtitle in a clear font with 4 options underneath, where one is the right answer. All displayed vividly with radio buttons next to them, only one radio button can be selected at a time.

The answer button is displayed at the bottom and that is used to verify your answer and determine whether it is correct or not, when clicking on the answer button a message box will appear displaying whether the question is right or wrong.

The 'Next' button is displayed in large so that the user will know where to click in order to proceed to the next question.

Exit button still displayed



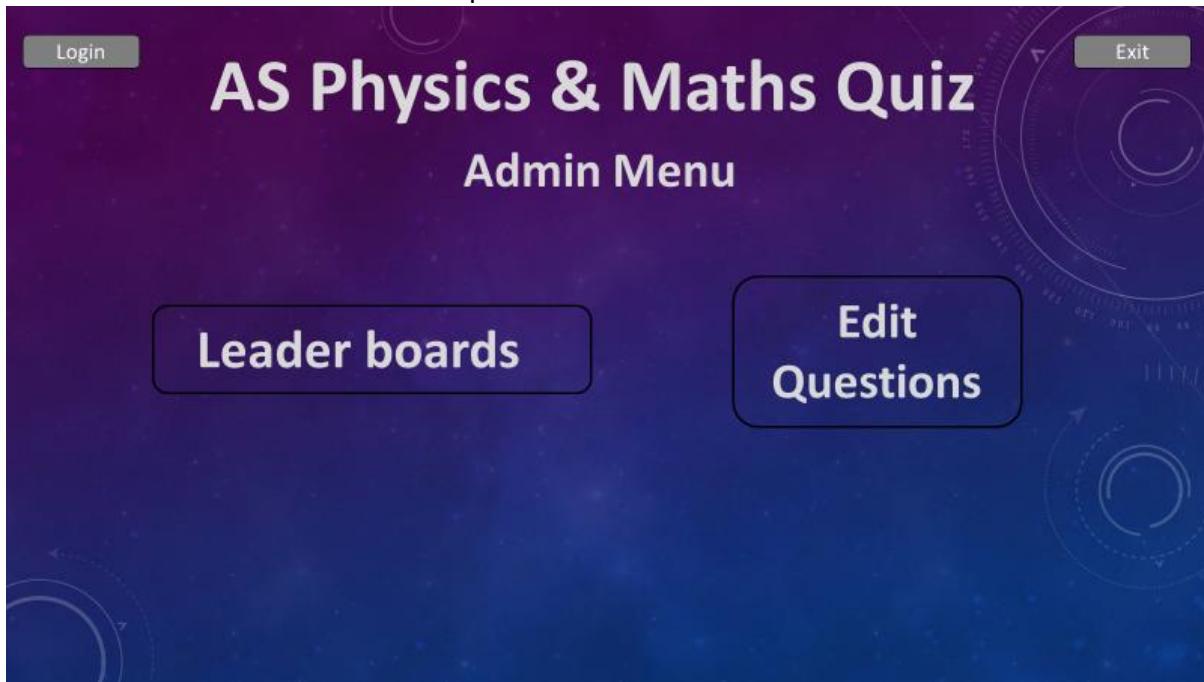
This is the feedback page, this will tell the user how many right answers they accumulated as well as their grade. The feedback is written in big bold font so that the feedback will be easy to see and read.

The quiz's title is still displayed at the top of the form so that the user knows which test they are obtaining feedback on.

The leader boards will display all of the results of each student in highest to lowest rank order, the button allows them to see their leader board and how they ranked compared to everyone else.

The 'Go Back To Main Menu' form directs the user back to the main menu where they can try another quiz, allow another user to login or to leave.

Exit button still displayed



This is the admin main menu, it is similar to the normal main menu yet the options are different. The title is still the same as well as the Exit and Login button.

The sub title informs the user that they are currently on the Admin Menu and not the Main Menu, this form will only be seen by admin accounts.

The button Leader Boards will direct the admin to all the leader boards for each subject so that they can see the progress of all their students and whether they need help or not.

The edit Questions button is essential to the menu, even though there are questions generated within the quiz, most of the questions will be based in a database that will be extracted. Therefore, the edit questions button will allow the admin user to add questions to the database as well as edit some questions already in there. That is the main reason for the admin accounts.

### **Functions:**

**Close:** The close function will run every time someone clicks on the 'exit' button. It will run a message box asking the user if they are sure that they want the program to end, if they click yes then the quiz will end if not then the message box will close, and the user will resume as normal.

```
Public Sub CloseBtn_Click( )
```

```
    End 'This button closes the whole program
```

```
End Sub
```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

**Next/Main Menu/Back To Login:** All these functions will do is load up a different windows user controls and then delete the current user control that is being displayed, this allows users to progress along the quiz.

'Sub routine which takes in user controls, and makes a choice on which uc to load

```
Public Sub addUC( )
```

```
newUC = New ucLogin 'Assigns a User Control to newUC so the variable isn't empty
```

```
Select Case ucChoice
```

```
Case Is = "ucLogin"
```

```
    newUC = New ucLogin 'Loads the user control uc Login
```

```
Case Is = "ucMainMenu"
```

```
    newUC = New ucMainMenu 'Loads the user control uc Main Menu
```

```
Case Is = "ucCreateAcc"
```

```
    newUC = New CreateAcc 'Loads the user control uc Create Account
```

```
Case Is = "MathsQuiz"
```

```
    newUC = New ucMathsQuiz 'Loads the user control uc Maths Quiz
```

```
Case Is = "ucPhysicsQuiz"
```

```
    newUC = New ucPhysicsQuiz 'Loads the user control uc Physics Quiz
```

```
Case Is = "ucAdminMenu"
```

```
    newUC = New ucAdminMenu 'Loads the user control uc Admin Menu
```

```
End Select
```

```
Me.Controls.Add(newUC) 'Adds the desired User Control
```

```
End Sub
```

**Login:** This function verifies that the username and password entered matches with the one in the database, this will then allow the user to enter the quiz with that login and all the results from tests will be saved to that login.

```
Private Sub LoginBtn_Click( )
```

Candidate Name: Naveed Ali Rafeeq Candidate Number: 1904  
dtAdmin = runSQL("Select \* from AdminData where TUserN = '" & UsrNmeTxt.Text & "' and TPass = '" & PassTxt.Text & "'")

dtStudent = runSQL("Select \* from StudentData where SUserN = '" & UsrNmeTxt.Text & "' and SPass = '" & PassTxt.Text & "'")

'Extracts all the Usernames and Passwords from the Admin and Student Tables

If dtStudent.Rows.Count = 1 Then

ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC

ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load

Me.Dispose() 'Gets rid of the user control

GC.Collect() 'Gather all the remaining data left after closing the user control

GlobalVariableUserName = dtStudent.Rows(0)(0) 'Saves that username to the global variable

Else

MsgBox("Your details have been entered incorrectly, please try again")

End If

End Sub

**Create New Account:** Opens the database where all the logins are placed and then allows the user to create a username and a password to be saved to that database so it can now be used for later use.

Private Sub NewAccBtn\_Click( )

dtCreateAcc = runSQL("Insert into StudentData (SUserN, SPass, TID) values ('" & NUsrNmeTxt.Text & "','" & NPassTxt.Text & "','" & TIDTxt.Text & "')")

MsgBox("New Account Added, you can now login")

End Sub

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

**Questions:** Each form within the test is a question therefore, as each form loads a random question will be extracted from the topic database as well as the multiple-choice answers stored with it.

dtLdQtn = runSQL("Select \* from MQuestions") 'Loads the MQuestion field from the MQuestions table in the access database

RowAmount = dtLdQtn.Rows.Count 'Counts the amount of rows in the table and assigns the integer to RowAmount

Randomize()

ranNum = Rnd() \* (RowAmount - 1) 'This generates a random number from how many rows there was

QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random question from the table

Rbtn1.Text = dtLdQtn.Rows(ranNum)(Answer1)

RBtn2.Text = dtLdQtn.Rows(ranNum)(Answer2)

RBtn3.Text = dtLdQtn.Rows(ranNum)(Answer3)

RBtn4.Text = dtLdQtn.Rows(ranNum)(Answer4)

'Loads all the answers to the corresponding

**Answer:** The answer button will open a message box and tell the user whether their option from the radio buttons is correct or not and thus add it to an accumulator that will tally all their correct scores.

If Rbtn1.Checked Then 'If the 1st radio button is ticked then

dtLdAns = runSQL("Select \* from MQuestions where MQA1 = '" & Rbtn1.Text & "'")  
'Loads from the database the rows which have the MQA1 value equal to the radio button text

If dtLdAns.Rows.Count > 0 Then 'If there is more than 0 rows left then it outputs a message saying they are correct

MsgBox("CORRECT!)

Counter += 1

Total += 1

Candidate Name: Naveed Ali Rafeeq  
Else

Candidate Number: 1904

MsgBox("INCORRECT!")

Total +=1

End If

End If

**Feedback:** This function will display a grade from a select statement from U to A, as well as a line of feedback corresponding to how well they did. This will be displayed as well as their accumulated value of questions they got right out of 20. E.g. 15/20.

Question Right Label = Global Variable Question Right **'The text now displays the value of the global variable (which is their score)**

Select Case Global Variable Question Right

Case Is > 17

Label = "A\*"

Label 2 = "You're A Genius"

Case Is = 16 Or 17

Label = "A"

Label 2 = "Pretty Good"

Case Is = 14 Or 15

Label = "B"

Label 2 = "Decent"

Case Is = 12 Or 13

Label = "C"

Label 2 = "You Passed"

Case Is = 10 Or 11

Candidate Name: Naveed Ali Rafeeq  
Label = "D"

Candidate Number: 1904

Label 2 = "Unlucky"

Case Is = 8 Or 9

Label = "E"

Label 2 = "Peak"

Case Is < 8

Label = "U"

Label 2 = "Unacceptable"

End Select

'This select statement displays a certain message and a grade depending on what number the global variable question right.

**Leader Board Sort:** All the results from the user's most recent scores will be ranked from highest to lowest order using the bubble sort algorithm.

'All leader board results are stored in a one dimensional array

Private Function LeaderBrdSort (LeaderBrdResults)

for all elements of LeaderBrdResults

if LeaderBrdResults [i] > LeaderBrdResults [i+1]

swap(LeaderBrdResults [i], LeaderBrdResults [i+1])

end if

end for

return list

End Function 'This function sorts the results from lowest to highest which when read from the leader board will be from bottom to top in the array.

**Leader Board:** Loads up a List View with a list of all the ranked results, with the username along with it.

Private Sub ScoreLoad()

Candidate Name: Naveed Ali Rafeeq  
Dim dtLeader As New DataTable

Candidate Number: 1904

```
dtLeader = runSQL("SELECT * from Results = "" & GlobalVariableUsername & "")")
```

```
For x = 0 To dtLeader.Rows.Count - 1
```

Dim items(1) As String 'Only two columns in the results table therefore the array is only a 1 x 2

```
items(0) = dtLeader.Rows(x)(0)
```

```
items(1) = dtLeader.Rows(x)(1)
```

'Fills each column with the username and score

```
Dim LineNew As New ListViewItem(items)
```

MathsListScore.Items.Add(LineNew) 'Adds a new line to the list view that is the username and results

```
Next
```

```
End Sub
```

**Add Questions:** A function that will be used by admin accounts to add new questions and their answers to the certain topic database where the admin wishes. (The one below will be for a Physics Question).

```
runSQL("INSERT into PQuestions (PQuestion, PA1, PA2, PA3, PA4) values ('" & Question TextBox & "," & TextBox 1 & "," & TextBox 2 & "," & TextBox 3 & "," & TextBox 4 & "')")
```

Adds all the values within the textboxes to each field of a row in the Physics Question Table

**Delete Questions:** Selects one item from the Listview row and then uses that string value to select all the items from that row in the question table and then deletes it. (The one below will be for the Maths Table).

```
runSQL("DELETE * FROM MQuestions WHERE MQuestion = '" & ListView.SelectedItems(0).SubItems(0).Text & "'")
```

The selected row in the List View that contains the question and answers is deleted from the Maths Question table

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

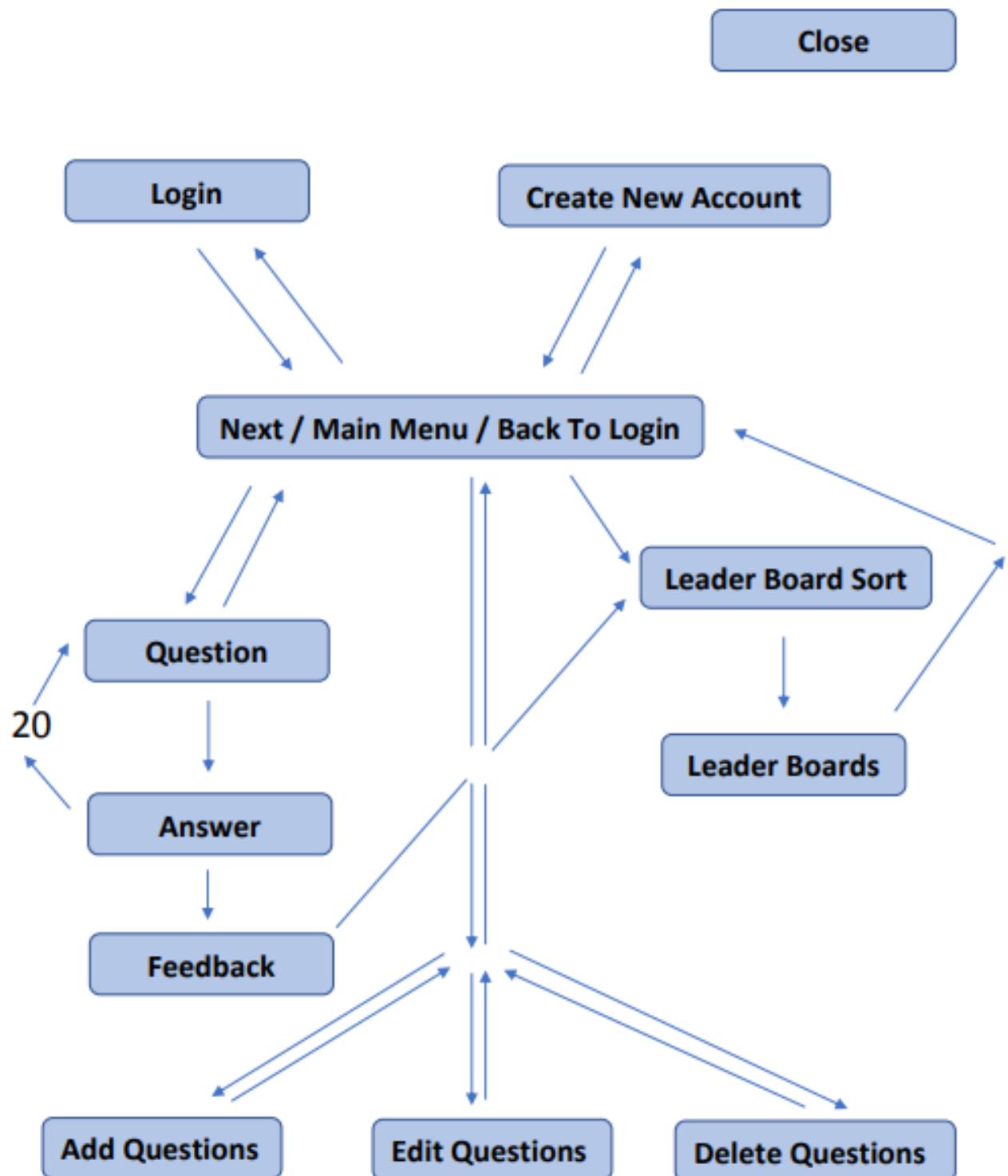
**Edit Questions:** A list will load with all of the questions will be loaded, then the question and each answer will be displayed into separate text boxes, the admin can then edit the question as well as the answers that come with it. (The one below will be for the Maths Table).

runSQL("DELETE \* FROM MQuestions WHERE MQuestion = '" & ListViewChoice

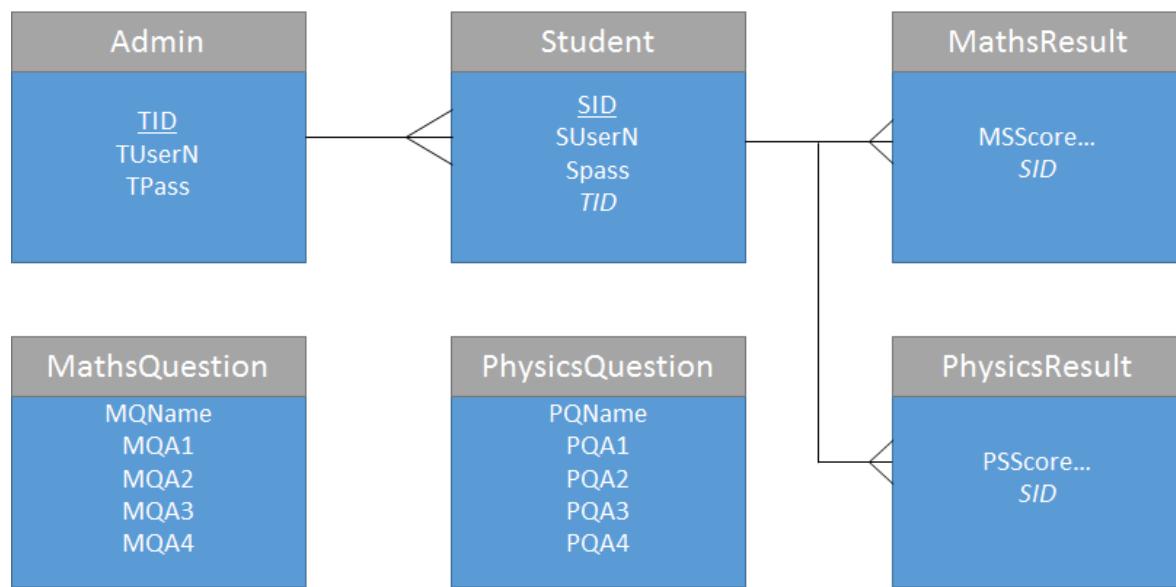
Deletes the whole row in the Maths Question table that contains the question selected

runSQL("INSERT into MQuestions (MQuestion, MA1, MA2, MA3, MA4) values ('" & Question  
Textbox & "," & Textbox 1 & "," & Textbox 2 & "," & Textbox 3 & "," & Textbox 4 & "")")

Adds the question and the 4 answers within the textboxes into the same table

**How The Functions Link Together:**

The Close function is apparent all throughout the program not linked to any other function therefore I placed it in the top right.

**Entity Relationship Diagram:****Data Dictionary:****Admin**

Field	Type	Description	Example
TID	Short Text	This is the primary key, given to each admin in the database.	TID3
TUserN	Short Text	The username of the admin that they will use to login.	BrendanFoster
TPass	Short Text	The corresponding username of the admin used to login as well.	ExamplePassword123

**Student**

Field	Type	Description	Example
SID	Short Text	This is the primary key, given to each student in the database.	SID4

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

SUserN	Short Text	The username of the student that they will use to login.	MoazMasood
SPass	Short Text	The corresponding username of the student used to login as well.	ExamplePassword123
TID	Short Text	This foreign key creates the relationship between the Admin and his students.	TID2

**MathsResult**

Field	Type	Description	Example
PSScore...	Number	This is the results of their physics tests; each test will add an extra field each time as well as a new field name of +1. E.g. PSScore3	14
SID	Short Text	This foreign key creates the relationship between the MathsResult and the specific student.	SID3

**PhysicsResult**

Field	Type	Description	Example
MSScore...	Number	This is the results of their maths tests; each test will add an extra field each time as well as a new field name of +1. E.g. MSScore3	9
SID	Short Text	This foreign key creates the relationship between the	SID3

		PhysicsResult and the specific student.	
--	--	-----------------------------------------	--

**MathsQuestion**

Field	Type	Description	Example
MQName	Short Text	This is the actual question that is displayed near top of the form above the multiple answers.	Find dy/dx of $y = 3X^2 + X$
MQA1	Short Text	This is one of the answers within the database and is always the correct answer. In the form it would be displayed randomly so the answers to all the questions will not be the first answer displayed.	$Dy/dx = 6X + 1$
MQA2	Short Text	Incorrect Answer to be displayed	$Dy/dx = 6X$
MQA3	Short Text	Incorrect Answer to be displayed	$Dy/dx = 3X + 1$
MQA4	Short Text	Incorrect Answer to be displayed	$Dy/dx = 6X^2 + X$

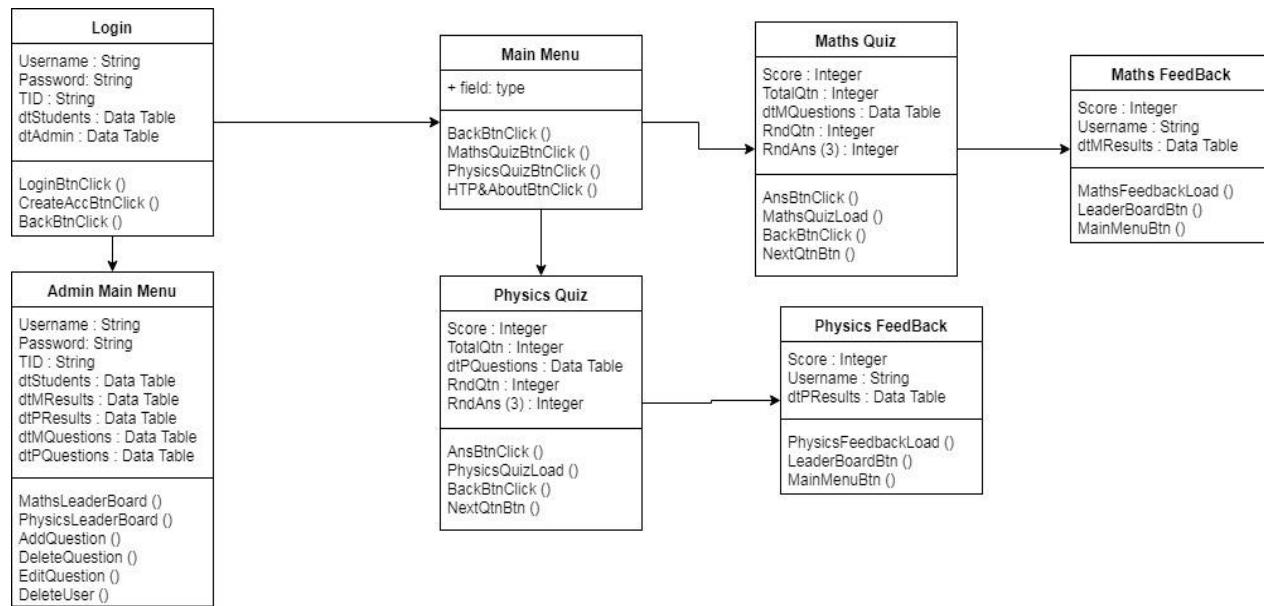
**PhysicsQuestion**

Field	Type	Description	Example
PQName	Short Text	This is the actual question that is displayed near top of the form above the multiple answers.	What is the equation for the frequency of a wave?
PQA1	Short Text	This is one of the answers within the database and is always the correct answer. In the form it would be displayed randomly so	$F = c / \lambda$

		the answers to all the questions will not be the first answer displayed.	
PQA2	Short Text	Incorrect Answer to be displayed	$F = 1/t$
PQA3	Short Text	Incorrect Answer to be displayed	$F = \lambda / c$
PQA4	Short Text	Incorrect Answer to be displayed	$F = ma$

## Classes:

I used a UML Class diagram to display all of my classes/forms, this universal method allows others to understand how my program works and how you get to each part of it.



## Variables:

All variables are Local unless stated otherwise

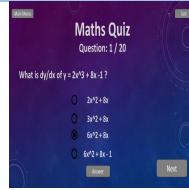
Variable Name	Data Type	Purpose	Sample Data	Scope
User_name	String	Used to check if the	MoazMasood	Login

		variable input is the same as the username within the database.		form
User_pass	String	Used to check if the variable input is the same as the password within the database.	Password786	Login form
New_user	String	The new account name that has just been created and is to be stored in the database.	ShaneWalters	Login form
New_pass	String	The new password for the new account that has just been created and is to be stored in the database.	Pass123	Login form
User_score	Integer	The total amount of questions that the user gets right.	15	Quiz forms
Main_choice	NA	Runs whichever form that links to the option the user clicked on in the main menu.	Select Case	Menu form
Q_choice	Boolean	If correct increment 'User_score'.	If Q_choice = database (QA1)	Quiz forms
dt	DataTable	Extracts a table from the database and then duplicates it within visual basic.	NA	Global Variable
da	DataAdapter	Establishes a link between the database	OleDb.OleDbDataAdapter	Global Variable

		and the program.		e
dtAdmin	DataTable	Creates a Table that replicates the Admin table within the database.	NA	Login form
dtStudent	DataTable	Creates a Table that replicates the Student table within the database.	NA	Login form
newUC	UserControl	An IF statement that loads different User Controls depending on what's called upon.	ucMainMenu	Main form
ucLoad	DataTable	Sets ucLoad to the parent of main form which allows ucLoad to access AddUC	Me.Parent	In all of the forms
FRanNum	Integer	Creates a random number between 1 and 4	3	Maths Quiz form
dtLoadQuestions	DataTable	Creates a Table that replicates the MQuestions table within the database.	NA	Maths Quiz form
RowAmount	Integer	Counts the number of rows in the	5	Maths Quiz

		dtLoadQuestions Table		form
ranNum	Integer	Generates a random number from the decremented value of RowAmount	2	Maths Quiz form
RanArray	Array(Integers )	A one-dimensional array that has values between 1-4 in it.	1	Maths Quiz form
Temp	String	Used as a temporary storage variable outside of the array for the shuffle algorithm.	3	Maths Quiz form
Value1	Integer	Chooses a value at random in the array	3	Maths Quiz form
Value2	Integer	Chooses another value at random in the array	4	Maths Quiz form
dtLoadAnswers	DataTable	Creates a table of only the correct answer column	NA	Maths Quiz form
dtLeader	DataTable	Creates a table with the usernames and their score from the database	NA	Quiz forms

**Validation:**

Validation Type	Location	Reason	Example
Radio Buttons	Maths & Physics question forms.	This will therefore only allow the user to pick one answer and not all of them. (Like with a tick box)	
Username check	When creating a new account in the login form.	This will check the database if there is already a user with that name and therefore it will ask you to choose another username.	User1
Length check	When creating a new account in the login form.	This will make sure that the password of the new user will be a minimum amount of characters long.	Password
Character check	When creating a new account in the login form.	So that the password created by the user has at least one number within it as well as letters.	Password123
Password and confirm password comparison check.	When creating a new account in the login form.	So that the user may not have inputted the wrong password by mistake.	Password786
Teacher ID check	When creating a new account in the login form.	This will make sure that a real teacher ID is used.	TID3

**Test Data For Development:**

LMC = Left Mouse Click

RMC = Right Mouse Click

**Login Form:**

Test Data	Type Of Test Data
Enter all letters for username	Valid
Enter all numbers for username/password	Invalid
Enter symbols for username/password	Invalid
Enter numbers and letters for username/password	Valid
Enter username/password less than 6 characters	Invalid - Extreme
Enter username/password 6 characters long	Valid - Extreme
Enter 0 characters for username/password	Invalid
Enter username/password 9 characters long	Valid
Press the enter button when in the Teacher ID textbox	Valid
Enter 2 different passwords in the password and confirm password boxes	Invalid
Enter a Teacher ID not in the Admin table	Invalid
Enter a username already in use	Invalid

**Navigation Through Forms:**

Test Data	Type Of Test Data
-----------	-------------------

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

LMC on Exit button	Valid
RMC on buttons	Invalid
LMC not on the buttons	Invalid
LMC on Back button	Valid

**Main Menu:**

Test Data	Type Of Test Data
LMC on Maths Quiz button	Valid
LMC on Physics Quiz button	Valid
LMC on How To Play / About button	Valid
Click where no buttons	Invalid
RMC on Maths Quiz button	Invalid
RMC on Physics Quiz button	Invalid
RMC on How To Play / About button	Invalid

**Question Form:**

Test Data	Type Of Test Data
LMC on a single radio button	Valid
LMC on 2 radio buttons at 1 time	Invalid
LMC Answer button	Valid
RMC Answer button	Invalid
RMC any radio buttons	Invalid
LMC on Next button	Valid

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

RMC on Next button	Invalid
--------------------	---------

### Feedback Form:

Test Data	Type Of Test Data
Score 14	Valid
Score 25	Invalid
Score 20	Valid – Extreme
You got an A	Valid
You got a Z	Invalid
Score -1	Invalid
LMC on score	Invalid – Erroneous
Score 21	Invalid – Extreme
LMC on Leader boards button	Valid

### Admin Menu Form:

Test Data	Type Of Test Data
LMC on Leader Boards button	Valid
RMC on Leader Boards button	Invalid
LMC on Edit Questions button	Valid
RMC on Edit Questions button	Invalid
Keyboard	Invalid – Erroneous

### Test Data For Beta Testing:

**Logging In & Creating Users:**

<b>Test Number</b>	<b>What is being Tested?</b>	<b>Input/Action</b>	<b>Expected Outcome</b>
1	Saves the users username and password to database.	<b>Valid:</b> Create new account button is pressed and a username and password are created.	When the user logs in with this username and password it should work.
2	Allows the User to continue onto the main menu form if their login matches one in the database.	<b>Valid:</b> Input the login details of an already created student or teacher account.  <b>Invalid:</b> Input incorrect details into the username and password textboxes and click login.	<b>Valid:</b> When the user presses the enter button on password or click the login button it should direct them to the Main Menu.  <b>Invalid:</b> A message box should appear telling the user that their details are incorrect.
3	When creating a new user, it checks whether the password and the confirm password textboxes are the same.	<b>Valid:</b> Both the textboxes have the same case-sensitive characters.  <b>Invalid extreme:</b> Both the text boxes have the same characters but not all of them are in the same case.	<b>Valid:</b> The program carries on the create user sub as usual.  <b>Invalid:</b> A message box informing the users that the passwords are wrong and then exit the sub.
4	When creating a new user, it checks whether the	<b>Valid:</b> The characters within the textbox is not in the Students or Teachers	<b>Valid:</b> The program carries on the create user sub as usual.

	username within the username textbox has already been taken.	tables.  <b>Invalid:</b> The characters are already in one of the 2 tables.	<b>Invalid:</b> A message box informs the users that the username has already been taken and then exits the sub.
5	When creating a new user, it checks whether the Teacher ID they inputted within the textbox matches one within the database.	<b>Valid:</b> The characters within the TID text box matches with one within the Teachers table.  <b>Invalid:</b> The characters within the TID text box does not match with one within the Teachers table.	<b>Valid:</b> The program creates the account within the student table and therefore they can now login.  <b>Invalid:</b> A message box informs the users that the teacher ID is not valid and then exits the sub.
6	How to log in into the admin main menu.	<b>Valid:</b> Click on the log in button with the correct teacher details.  <b>Invalid:</b> Click on the log in button with correct student details.	<b>Valid:</b> The admin main menu form should load.  <b>Invalid:</b> The main menu form should load.

**Navigation Between Forms:**

Test Number	What is being Tested?	Input/Action	Expected Outcome

7	When logging in, the user can jump into the main menu without having to press the log in button.	<p><b>Valid:</b> The user presses the enter button in the password textbox.</p> <p><b>Invalid Extreme:</b> The user presses the enter button in the password textbox but with wrong details.</p>	<p><b>Valid:</b> The main menu screen logs in if the user's login details are correct.</p> <p><b>Invalid extreme:</b> A message box appears informing the user that the details are incorrect.</p>
8	How to exit the program.	<p><b>Valid:</b> The user clicks on the "Exit" button in the top right of every form.</p> <p><b>Invalid Erroneous:</b> The user clicks on the form background.</p>	<p><b>Valid:</b> A message box appears asking if they are sure they want to leave.</p> <p><b>Invalid:</b> Nothing happens.</p>
9	How to go back to the login form from the main menu / create account form.	<p><b>Valid:</b> Click the "Login" button in the top left of the main menu and the create account forms</p> <p><b>Invalid:</b> Click the Exit button in the top right of the main menu / create account forms.</p>	<p><b>Valid:</b> The main menu / create account form closes and the login form loads.</p> <p><b>Invalid:</b> A message box loads asking if they are sure they want to leave.</p>
10	How to exit a quiz but stay on the program.	<p><b>Valid:</b> Click on the back button in the top left of the quiz, and then click yes on the message box that appears.</p>	<p><b>Valid:</b> The main menu form should load on the program.</p>

		<b>Invalid Extreme:</b> Click on the back button in the top left of the quiz, and then click NO on the message box that appears.	<b>Invalid Extreme:</b> It should return you to your quiz and the same question that the user was currently on.
11	How to start a maths / physics quiz.	<b>Valid:</b> Whilst on the main menu form click on the maths / physics quiz button.	<b>Valid:</b> The corresponding quiz should load on the program with the subtitle of Question: 1 / 20.
		<b>Invalid:</b> Whilst on the main menu form click on the about button.	<b>Invalid:</b> The about form should load with the details of the quiz.
12	How to view the leader boards as a student.	<b>Valid:</b> Click on the view leader boards button on the feedback form	<b>Valid:</b> A list view of the students ranked by scores should load.
		<b>Invalid:</b> Click on the back to main menu button on the feedback form.	<b>Invalid:</b> The main menu should load.
13	How to view the leader boards as a teacher.	<b>Valid:</b> Click on the leader boards button on the admin menu form.	<b>Valid:</b> A list view of the students ranked by scores should load.
		<b>Invalid:</b> Click on the back to login button in the top left of the admin menu form.	<b>Invalid:</b> The teacher should be signed out and the login form should load.

14	How to go to the next question in the quizzes.	<b>Valid:</b> Click on the next button after clicking the answer button.  <b>Invalid Extreme:</b> Click on the next button before clicking on the answer button.	<b>Valid:</b> The next question will load on the same form and the question number will become incremented.  <b>Invalid Extreme:</b> Nothing will happen due to the answer button not being clicked first.
----	------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Maths / Physics Quiz:**

Test Number	What is being Tested?	Input/Action	Expected Outcome
15	Answering a question after clicking on one of the four radio buttons.	<b>Valid:</b> Click on the answer button.	<b>Valid:</b> A message box will load.
		<b>Invalid Erroneous:</b> Press the enter key	<b>Invalid Erroneous:</b> Nothing happens.
16	Answering a question without clicking on one of the four radio buttons first.	<b>Valid:</b> Click on the answer button.	<b>Valid:</b> Nothing happens.
		<b>Invalid Erroneous:</b> Press the right arrow key.	<b>Invalid Erroneous:</b> Nothing happens
17	Choosing a correct answer.	<b>Valid:</b> Click the right radio button corresponding to the question and then	<b>Valid:</b> A message box should load with the text correct, and then the

		click the answer button.	right answer and question counter should both increment.  <b>Invalid:</b> Nothing happens.
18	Choosing an incorrect answer	<b>Valid:</b> Click the wrong radio button corresponding to the question and then click the answer button.  <b>Invalid Erroneous:</b> Press the enter button after choosing a radio button.	<b>Valid:</b> A message box should load with the text incorrect, and then the question counter should increment.  <b>Invalid:</b> Nothing happens.
19	The correct title loads with its corresponding quiz Maths / Physics.	<b>Valid:</b> Click on a quiz in the main menu.  <b>Invalid Extreme:</b> Click on the about button.	<b>Valid:</b> Maths / Physics quiz appears at the top of the program.  <b>Invalid:</b> About appears on the top of the program.
20	How to complete the quiz.	<b>Valid:</b> Answer 20 questions.  <b>Invalid Extreme:</b> Answer 19 questions.	<b>Valid:</b> The feedback form of the program will load.  <b>Invalid:</b> The last question of the quiz will be displayed.

<b>21</b>	Question structure	Complete the quiz	No question should be repeated.
<b>22</b>	Answer structure	Complete the quiz	All the answers to a question should be different to each other.
<b>23</b>	The question number should be reset when loading the quiz.	<b>Valid:</b> Every time the main menu is loaded.  <b>Invalid:</b> Click on main menu button in the top right and then click no on the message box.	<b>Valid:</b> The question counter returns to 0.  <b>Invalid:</b> The question counter stays unchanged therefore so does the question number.
<b>24</b>	The right answer counter should reset when leaving the quiz.	<b>Valid:</b> Every time the main menu is loaded.  <b>Invalid:</b> The exit button is clicked and then the user clicks yes on the message box.	<b>Valid:</b> The right answer counter returns to 0.  <b>Invalid:</b> The program closes.
<b>25</b>	An even number of times a question with random values appears.	Complete the quiz	Roughly half of the questions are from the random number generated.

**Feedback & Leader Boards:**

Test Number	What is being Tested?	Input/Action	Expected Outcome

26	The leader boards are ranked by the scores of each student in a class.	<p><b>Valid:</b> Click on the leader boards button.</p> <p><b>Invalid:</b> Click on the main menu button in the bottom right of the feedback form.</p>	<p><b>Valid:</b> A sort algorithm should sort out the order of the scores in a list view</p> <p><b>Invalid:</b> The main menu form should load.</p>
27	The values in the leader board cannot be changed.	<p><b>Valid:</b> Click on a user and type in characters on the keyboard.</p> <p><b>Invalid:</b> Click on the main menu button in the bottom right.</p>	<p><b>Valid:</b> Nothing happens.</p> <p><b>Invalid:</b> The main menu form loads.</p>

### Editing Questions & Accounts

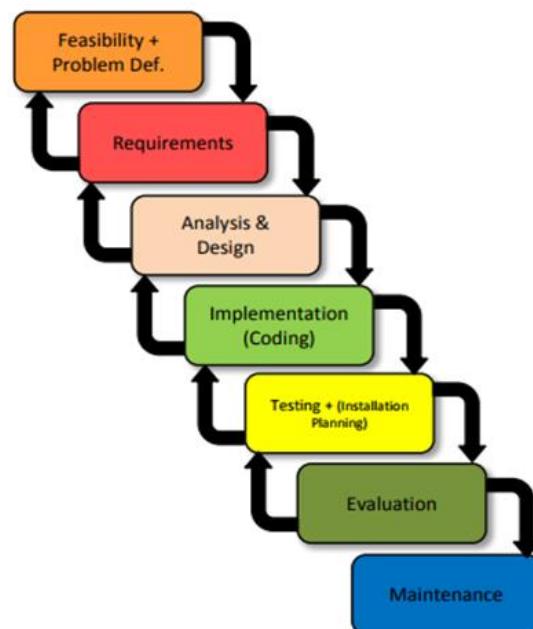
Test Number	What is being Tested?	Input/Action	Expected Outcome
28	The teacher can add questions to the quiz.	<p><b>Valid:</b> The teacher types a question and four answers into separate textboxes and clicks add question.</p> <p><b>Invalid Extreme:</b> The teacher clicks the add question button without typing into the textboxes.</p>	<p><b>Valid:</b> The new question is added to the M/P Questions table.</p> <p><b>Invalid:</b> A message box appears informing the teacher that the textboxes are blank.</p>

29	The teacher can't add the same question twice.	<b>Valid:</b> The teacher adds a question that's already in the M/P Questions table.  <b>Invalid:</b> The teacher adds a question already in the physics quiz into the maths quiz or vice versa	<b>Valid:</b> A message box appears saying the question has already been added.  <b>Invalid:</b> The question is added.
30	The teacher can delete an account from their class.	<b>Valid:</b> The teacher clicks on the accounts button and then clicks on the user they wish to delete, then click the delete button.  <b>Invalid Extreme:</b> The teacher clicks on the accounts button and then clicks on the user they wish to delete.	<b>Valid:</b> The user is now deleted from the Students table in the database, they can no longer log in.  <b>Invalid Extreme:</b> Nothing happens.

## SDLC Process:

### Waterfall Model:

This lifecycle is well known, it consists of a sequence of stages and each stage is only started after previous one is complete. However, you can go back a stage when necessary. This system worked really well with the start of my project since there is a clear expected output at the requirements, analysis and design stages, but once I got to the implementation stage the outcome wasn't so



**Candidate Name:** Naveed Ali Rafeeq

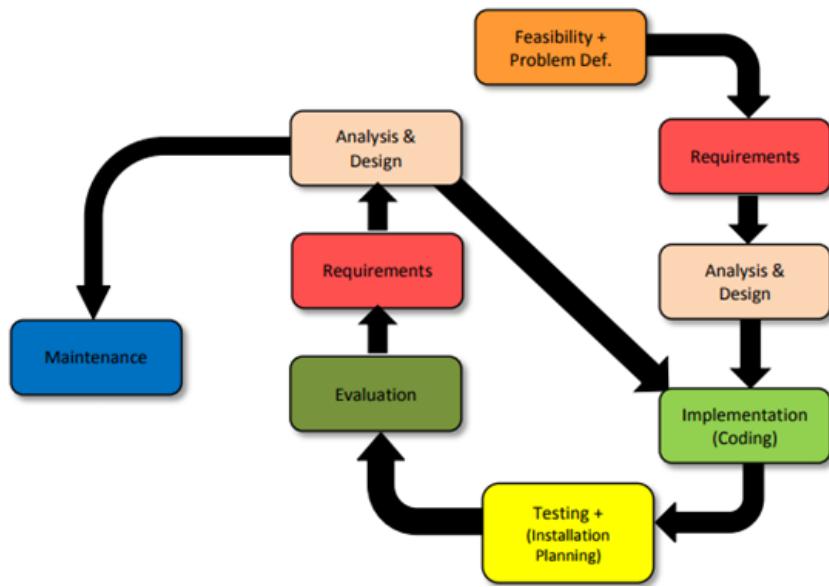
Candidate Number: 1904

clear and I had to keep testing as well therefore making the Waterfall lifecycle not as efficient.

## Agile Methodology:

The agile lifecycle is a group of methods from different methodologies. This method understands that the requirements change throughout producing the software and therefore deals with this by iterating and therefore produces different versions, each with additions on the previous versions and each with more requirements it has to meet. The lifecycle ends when either all of the requirements have been met or when the time left to develop the software has ran out. I will be using this method for the most part of my project, I will produce different prototypes of each software each building upon the previous. With each prototype I will interview my stakeholders and update my system requirements with each version.

```
graph TD; A[Feasibility + Problem Def.] --> B[Requirements]; B --> C[Analysis & Design]; C --> D[Implementation<br>(Coding)]; D --> E[Testing +<br>(Installation Planning)]; E --> F[Evaluation]; F --> B; B --> G[Maintenance];
```

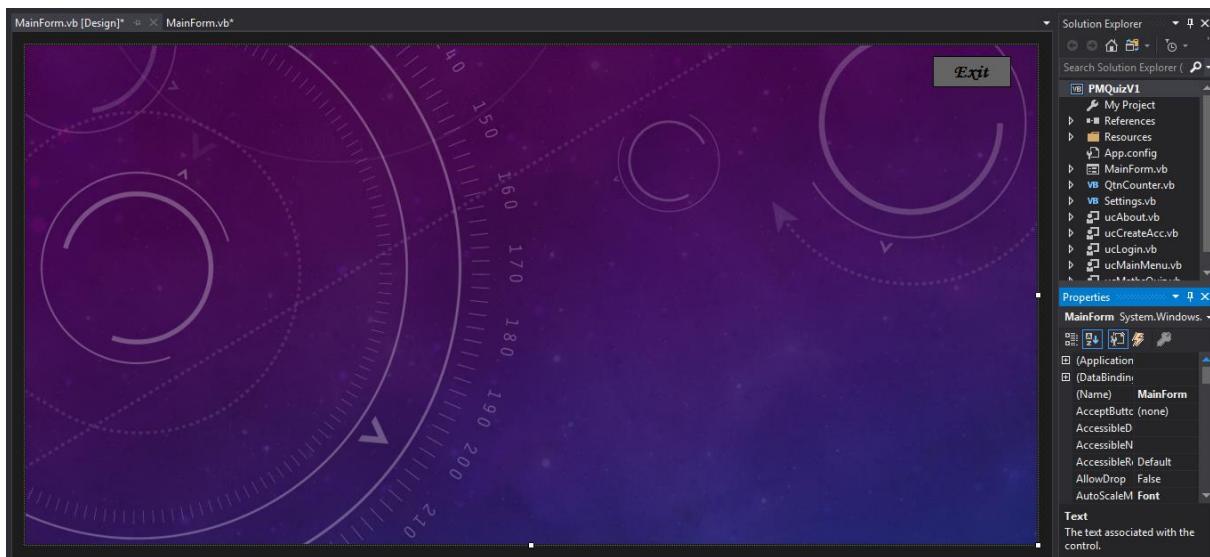


## C. DEVELOPING THE CODED SOLUTION ("THE DEVELOPMENT STORY")

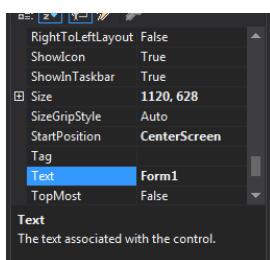
### Prototype 1

Even though I knew what my requirements were, I still followed the iterative agile methodology therefore producing three prototypes. With each prototype I interviewed my stakeholder Brendan Foster in order for him to inform me on what needed improvement within my program. Before starting my development, I looked at some of my Visual Basic code from last year since I was able to copy some of it for my program. My whole login and create user system is based off that since it was very similar to what I needed to create. At first, I was going to use forms for each page of my quiz but then I decided to use User Controls instead (reason will be explained in my evaluation). The final prototype will end once I either run out of time for my project or all of the system requirements have been met.

#### Creating the Form for my game:

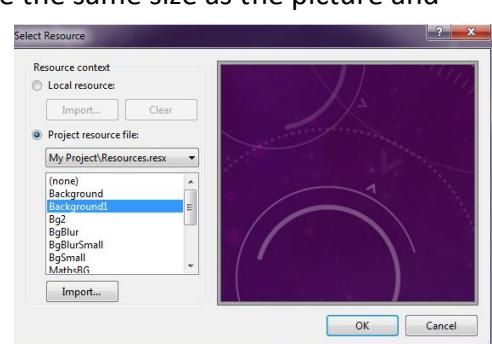


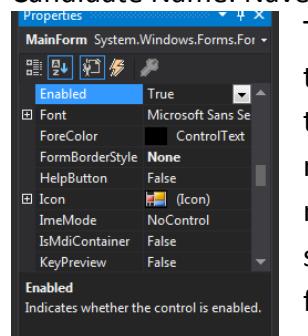
The first thing I did was create a windows form that had a background and a button, my intention is for these two to stay constant throughout.



I decided that the form should be the same size as the picture and therefore made it 1120 by 628 pixels.

I selected the background from my resources that are saved onto visual basic.





Then I changed the border style to none this therefore does not allow the user to resize the program, it also does not allow the user to close the program so therefore I added the exit button on the form. I decided not to make my program full screen since there are many different sized monitors thus causing problems with each monitor used, so I choose that size 1120 by 628 pixels due to it being a smaller size than the full screen for most monitors.

```

30     Public Sub CloseBtn_Click(sender As Object, e As EventArgs) Handles CloseBtn.Click
31
32         End 'This button closes the whole program
33
34     End Sub
35 End Class

```

The exit button runs this sub routine when it is clicked which therefore terminates the program.

### Creating The User Controls:

So instead of creating a lot of code for each user control I instead used a select case statement in order to condense the code.

```

Public Class MainForm
    Private Sub MainForm_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        addUC("ucLogin")
        'Places the User Control Login when the mainform loads
    End Sub

    'Sub routine which takes in user controls, and makes a choice on which uc to load
    Public Sub addUC(ByVal ucChoice As String)
        Dim newUC As UserControl 'variable newUC that adds new user control
        newUC = New ucLogin 'Assigns a User Control to newUC so the variable isn't empty

        Select Case ucChoice
            Case Is = "ucLogin"
                newUC = New ucLogin 'Loads the user control ucLogin
            Case Is = "ucCreateAcc"
                newUC = New CreateAccBtn 'Loads the user control ucCreateAcc
        End Select
        Me.Controls.Add(newUC) 'Adds the desired User Control
    End Sub

```

I created two user controls from the project section and named them “ucLogin” and “CreateAccBtn”, and created a variable named newUC as a user control. I placed the user controls into a select case statement, so when the string of the user control is added to the statement it assigns the newUC user control to it. The addUC(“ucLogin”) is added so that when the program loads there is a user control to begin with.

```

Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
    Dim ucLoad As MainForm

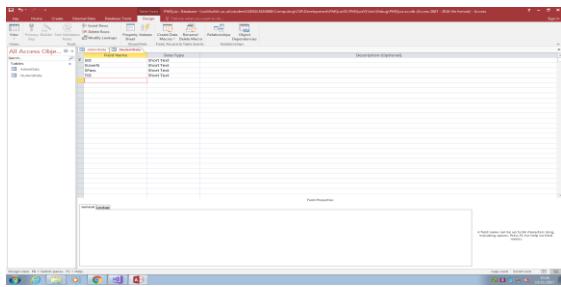
    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

```

So now in each user control I can add this code and it will remove the current user control and add this one instead.

## Creating my Database:



I created this database in Access so that the login can extract data from it when logging in, also when I go on to creating my create account User Control It will add data to this database. The AdminData table is also very similar however; those accounts will have to be entered manually since it they need their own bespoke Teacher ID.

## Establishing the connection to the database:

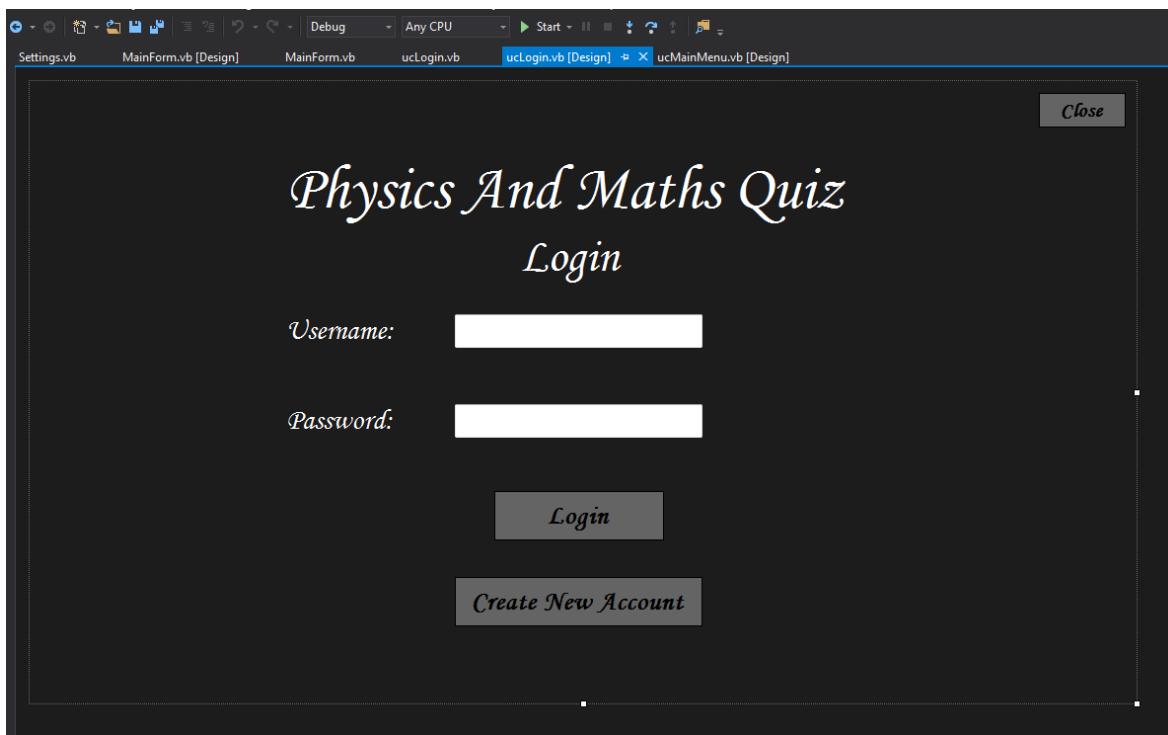
```

Public Module Settings

    'The module opens the database and creates a datatable in vb, then any edits that occur during the program then
    'it updates the database With the datatable
    Public Function runSQL(ByVal query As String) As DataTable
        Dim con As New OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=dbPMQuiz.mdb")
        'Establishes a connection to a access file by the name dbPMQuiz
        Dim da As OleDb.OleDbDataAdapter
        'This is the data adapter, this fetches data from the database and then translates between the database and VB
        'But it does not completely exist at this point
        Dim dt As New DataTable
        'Create the data table variable assigned to dt
        con.Open() 'Opens the database connection
        da = New OleDb.OleDbDataAdapter(query, con) 'The data adapter now exists and it runs the following query
        'by the connection
        dt.Clear() 'The table is cleared incase it already had values within it
        dt.Columns.Clear() 'Verification that the table is cleared
        da.Fill(dt) 'The data adapter is now filled with the database
        con.Close() 'Closes the connection to the database
        'It is important to close the connection because only one connection can be made at one time
        Return dt 'The data table filled with the database is now outputted from the function
    End Function
End Module

```

I created a public module called settings. When the program loads it automatically runs this module so that a connection to the database will be made straight away. It also condenses my code in each user control since I do not need to open and close a connection every time, which can become very tedious and cause more problems if I forget to close it.

**User Control – Login:**

The size of all my user controls are the same as my form since it needs to fit on it. The background is transparent because the form background will appear. (Please note that throughout my program I keep switching between exit and close for my top right button).

```
Public Class ucLogin
    Dim ucLoad As MainForm

    'Does a check to see if a key has been pressed when the user is in the password textbox
    Private Sub PassTxt_PreviewKeyDown(sender As Object, e As PreviewKeyDownEventArgs) Handles PassTxt.PreviewKeyDown
        If e.KeyCode = Keys.Enter Then
            Call LoginBtn_Click(sender, e)
        End If 'When the user presses enter button in the password textbox it runs the login button
    End Sub
```

This sub routine runs whenever the enter button is pressed within the password box, whenever the enter button is pressed it runs another subroutine within the class called LoginBtn\_Click (runs the login button). This makes logging into the quiz faster since the user doesn't need to click the actual button to run it. The ucLoad main form variable is created outside of any sub routine so it does not need to be repeated with each user control sub routine.

```

Private Sub LoginBtn_Click(sender As Object, e As EventArgs) Handles LoginBtn.Click
    Dim dtAdmin, dtStudent As DataTable 'Creates 2 datatable variables
    dtAdmin = runSQL("Select * from AdminData where TUserN = '" & UsrNmeTxt.Text & "' and TPass = '" & PassTxt.Text & "'")
    dtStudent = runSQL("Select * from StudentData where SUserN = '" & UsrNmeTxt.Text & "' and SPass = '" & PassTxt.Text & "'")
    'Extracts all the Usernames and Passwords from the Admin and Student Tables

    If dtAdmin.Rows.Count Or dtStudent.Rows.Count = 1 Then
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    Else
        MsgBox("Your details have been entered incorrectly, please try again")
    End If

End Sub

```

Whenever the user clicks the login button it runs this subroutine. Two data tables are created to duplicate and then eventually overrun the two tables (Admin and Student) within the dbPMQuiz database. It then runs the two SQL statements that extracts the exact values that are equal to the string values within the Username text box and the Password text box. Next, the IF statement counts the number of rows that are left within the data tables dtAdmin and dtStudent, if there is one row left then it loads the main menu user control. If the rows are not equal to one (should be 0) then it displays the message box informing the user that the details they entered where wrong and then does nothing.

```

Private Sub NewAccBtn_Click(sender As Object, e As EventArgs) Handles NewAccBtn.Click
    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.addUC("ucCreateAcc") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

```

**End Class**

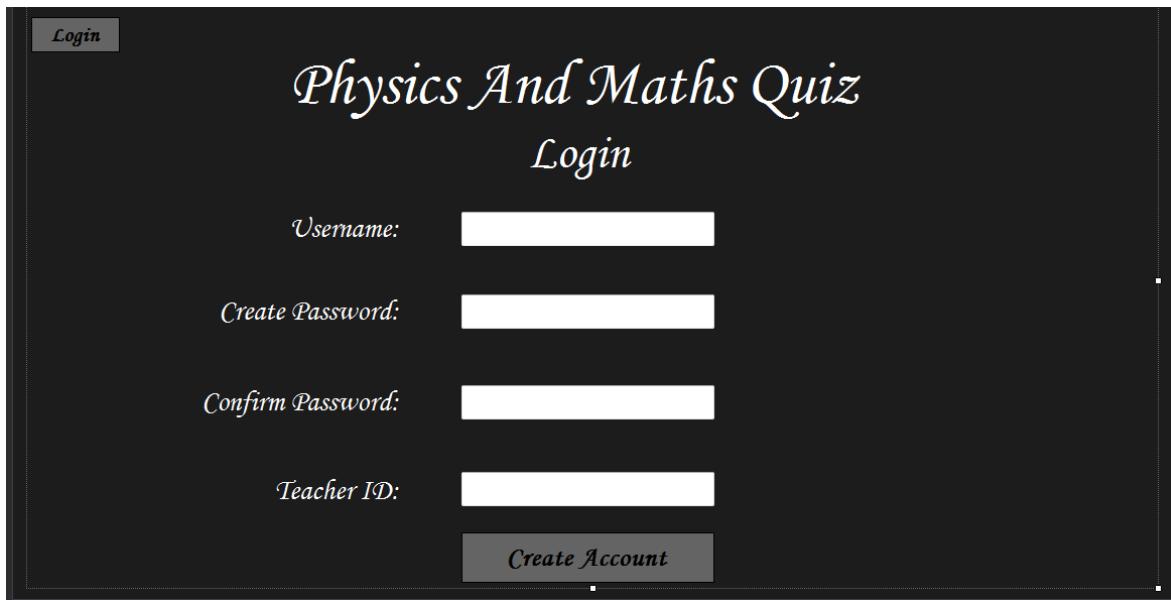
Finally, within the login user control, the user control Create Account is loaded if the Create new account button is clicked.



My User Control Main Menu so far.

1120 x 628 pixels.

Similar yet different background to the one used before.

**User Control – Create Account**

This user control consists of 4 textboxes and the two buttons create account and login.

```
'Does a check to see if a key has been pressed when the user is in the password textbox
Private Sub TIDTxt_KeyDown(sender As Object, e As KeyEventArgs) Handles TIDTxt.KeyDown
    If e.KeyCode = Keys.Enter Then
        Call NewAccBtn_Click(sender, e)
    End If 'When the user presses enter button in the password textbox it runs the login button
End Sub
```

As with before, when the enter key is pressed within the TID text box, it calls upon the NewAccBtn\_Click sub routine.

```
Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
    Dim ucLoad As MainForm

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control
End Sub
```

When the login button in the top left of the user control is clicked, the login user control is loaded and the create account user control disappears. The next sub is split into two parts.

```

Public Class CreateAccBtn
    Private Sub NewAccBtn_Click(sender As Object, e As EventArgs) Handles NewAccBtn.Click
        'All the if statements in this sub routine are forms of validation which stop the user from creating a new account

        Dim dtCreateAcc As DataTable 'Creates the table dtCreateAcc, a copy of the student table in the database

        If NPassTxt.Text <> NPass2Txt.Text Then
            MsgBox("Password do not match")
            Exit Sub
        End If
        'If the text in the NPass textbox and NPass2 textbox do not match then it exits the sub routine

        dtCreateAcc = runSQL("Select * from AdminData where TID = '" & TIDTxt.Text & "'")
        If dtCreateAcc.Rows.Count <> 1 Then
            MsgBox("Incorrect Teacher ID")
            Exit Sub
        End If
        'If the text in the TID textbox is not in the DataTable then it exits the sub routine
    End Sub

```

This sub acts as validation as well as inserting the user details into the database so that they can log in afterwards. If the create password textbox and the confirm password textbox string values are equal to each other the sub can then continue otherwise it will come up with an error message and then exit the sub so that the user's details are not inputted into the database.

Next it checks if the Teacher ID (TID) is an actual TID within the database. It does this by loading the whole AdminData table into a new Data Table; it then loads the rows that have a TID equal to the one within the textbox. Since there is only one of each TID, if there is not 1 row left in the Data Table then it outputs another error message and exits the sub, so no data can be entered into the database.

```

dtCreateAcc = runSQL("Select * from StudentData where SUserName = '" & NUserNameTxt.Text & "'")
If dtCreateAcc.Rows.Count > 0 Then
    MsgBox("Account name already in use, please use another")
    Exit Sub
    'If the text in the NUserName textbox is already in the SUserName column of the DataTable then it exits the sub routine
Else
    dtCreateAcc = runSQL("Insert into StudentData (SUserName, SPass, TID) values ('" & NUserNameTxt.Text &
                         "','" & NPassTxt.Text & "','" & TIDTxt.Text & "')")
    MsgBox("New Account Added, you can now login")
End If
'If it is not already in that column then it adds the text from the three text boxes into the student table in the database

Dim ucLoad As MainForm

ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which determines which user control to load
Me.Dispose() 'Gets rid of the user control
GC.Collect() 'Gather all the remaining data left after closing the user control
End Sub

```

The Data Table now contains the row that contains the username equal to the one within the username textbox, which should be nothing since the user is creating an account and therefore it should be a new username. However, if there is a row within the Data Table then the sub tells the user that that the account name has already been used and it exits the sub, so that no data can be entered into the database again.

If the username is not in the database the sub will then proceed to add the username, password and the TID into a new row of the database. Thus, a message informs the user that

Candidate Name: Naveed Ali Rafeeq Candidate Number: 1904  
 their details have been added. Next, the current UC (create account) will be disposed of and the UC Login will be loaded so that the user can now log in straight away.

## Testing Prototype 1

### Login & Create Account Form:

Test Data	Test	Type Of Test Data	Output
	Number		
Enter all letters for username	1	Valid	Saves to database
Enter numbers and letters for username/password	1	Valid	Saves to database
Enter 0 characters for username/password	2	Invalid	Saves to database
Enter username/password 9 characters long	1	Valid	Saves to database
Press the enter button when in the password textbox	NA	Valid	Runs the Login button sub
Press the enter button when in the Teacher ID textbox	NA	Valid	Runs the Create Account button sub
Enter 2 different passwords in the password and confirm password boxes	3	Invalid	Error message appears and exits sub
Enter a Teacher ID not in the Admin table	4	Invalid	Error message appears and exits sub
Enter a username already in use	5	Invalid	Error message appears and exits sub
Enter incorrect details and click the login	6	Invalid	Error message

The justification for the above testing is to make sure that the login and create account functions work. This is not only an integral part of the program, but it is also the first part of the program therefore by doing the above testing it will try to break the functions and therefore ensure that there will be no bugs within these functions throughout the prototypes.

#### **Navigation:**

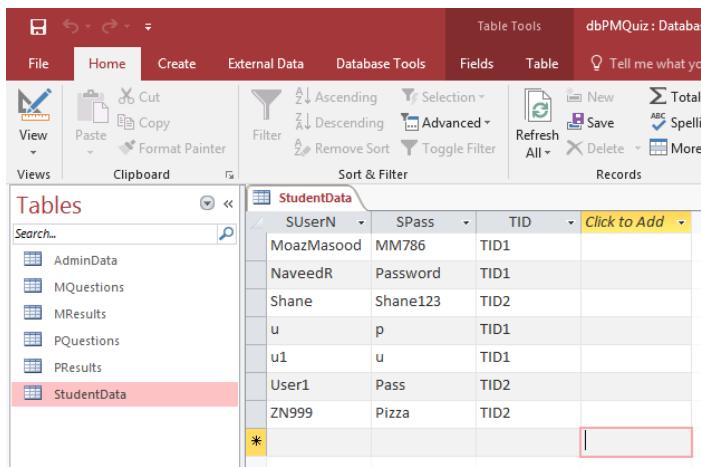
Test Data	Test Number	Type Of Test Data	Output
LMC on Exit button	7	Valid	Closes program
RMC on buttons	NA	Invalid	Nothing
LMC not on the buttons	NA	Invalid	Nothing
LMC on Login button in the Create Account UC	8	Valid	Message box appears then disposes current UC and loads the Login UC
Enter correct details and click on the Login button in the Login UC	9	Valid	Disposes current UC and loads the Main Menu UC

Not much testing is needed for the navigation throughout the quiz because of the user controls making the function of changing screens very simple and minimizes the amount of problems that could occur. Therefore as long as the testing above works then the navigation should work fine throughout the program.

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

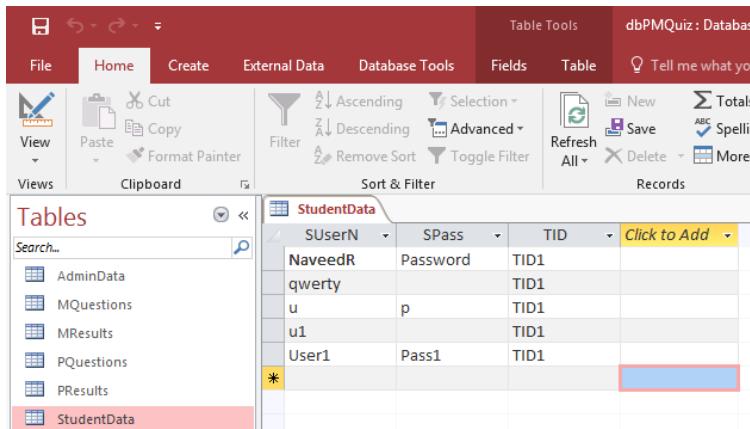
### Test 1



A screenshot of the Microsoft Access application interface. The ribbon at the top shows 'File', 'Home', 'Create', 'External Data', 'Database Tools', 'Fields', 'Table', and 'dbPMQuiz : Database'. The 'Home' tab is selected. On the left, the 'Tables' list shows 'StudentData' highlighted with a red box. The main area displays the 'StudentData' table with the following data:

SUserName	SPass	TID	Action
MoazMasood	MM786	TID1	
NaveedR	Password	TID1	
Shane	Shane123	TID2	
u	p	TID1	
u1	u	TID1	
User1	Pass	TID2	
ZN999	Pizza	TID2	

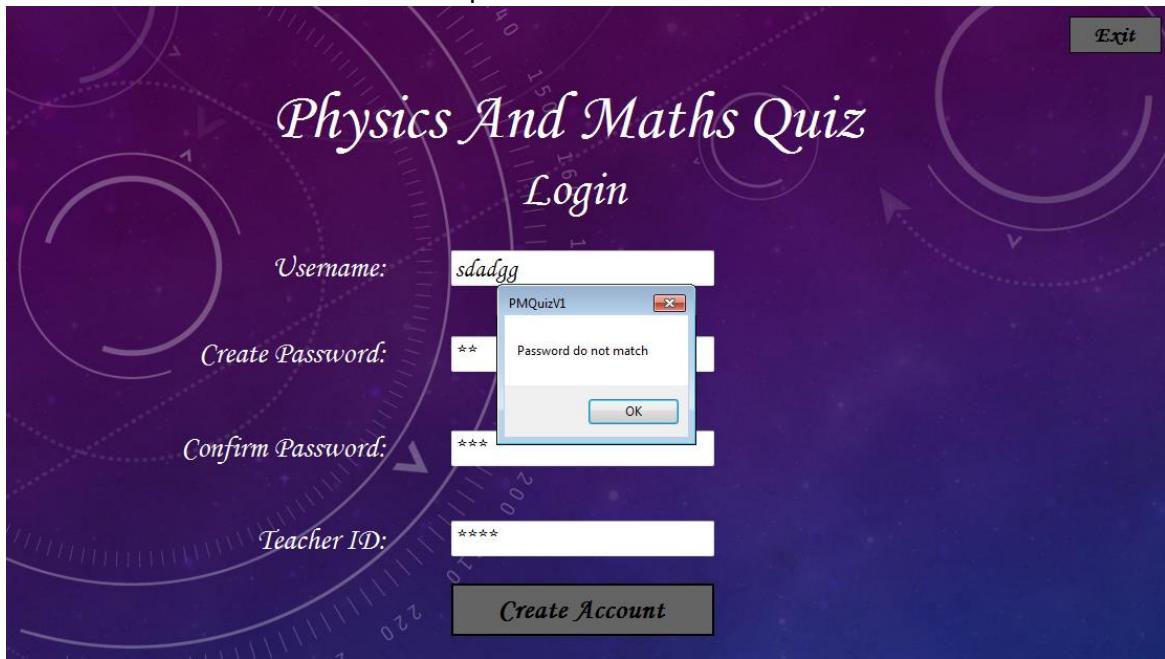
### Test 2



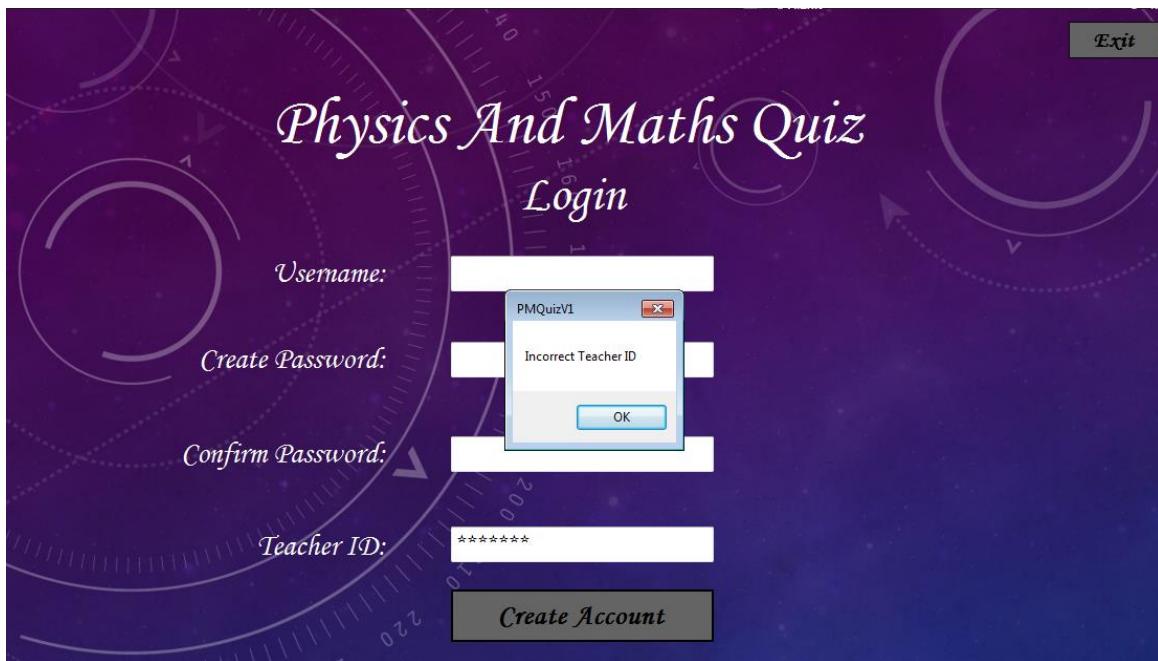
A screenshot of the Microsoft Access application interface, similar to Test 1. The 'Home' tab is selected. The 'Tables' list shows 'StudentData' highlighted with a red box. The main area displays the 'StudentData' table with the following data:

SUserName	SPass	TID	Action
NaveedR	Password	TID1	
qwerty		TID1	
u	p	TID1	
u1		TID1	

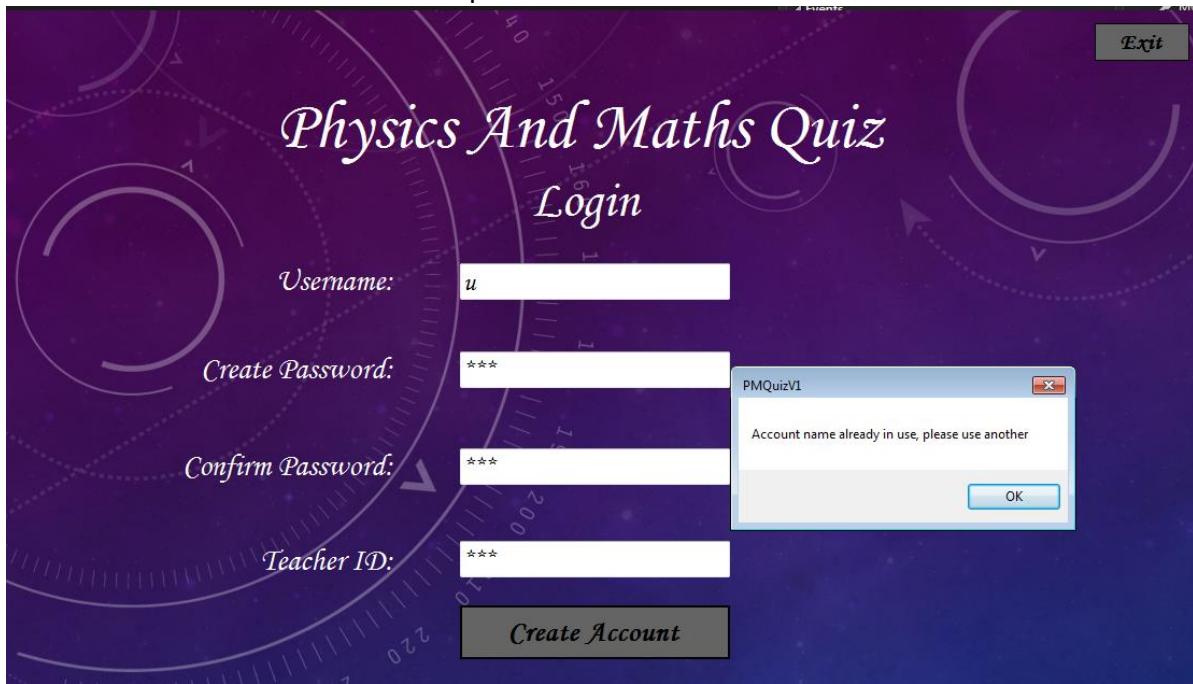
### Test 3



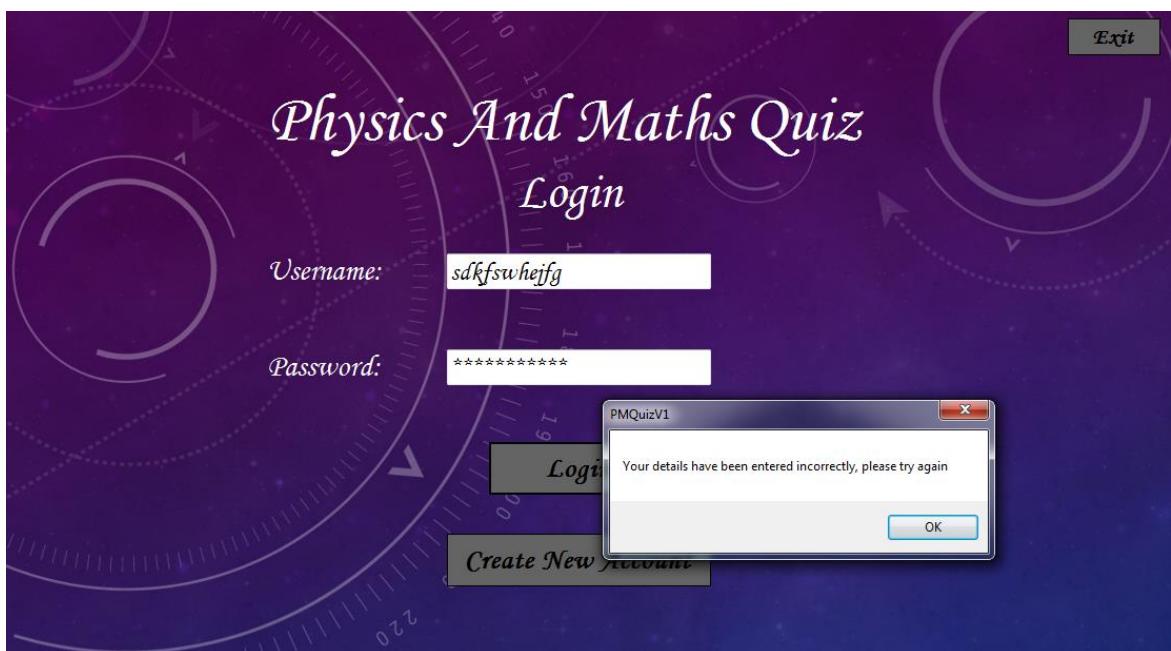
Test 4



Test 5



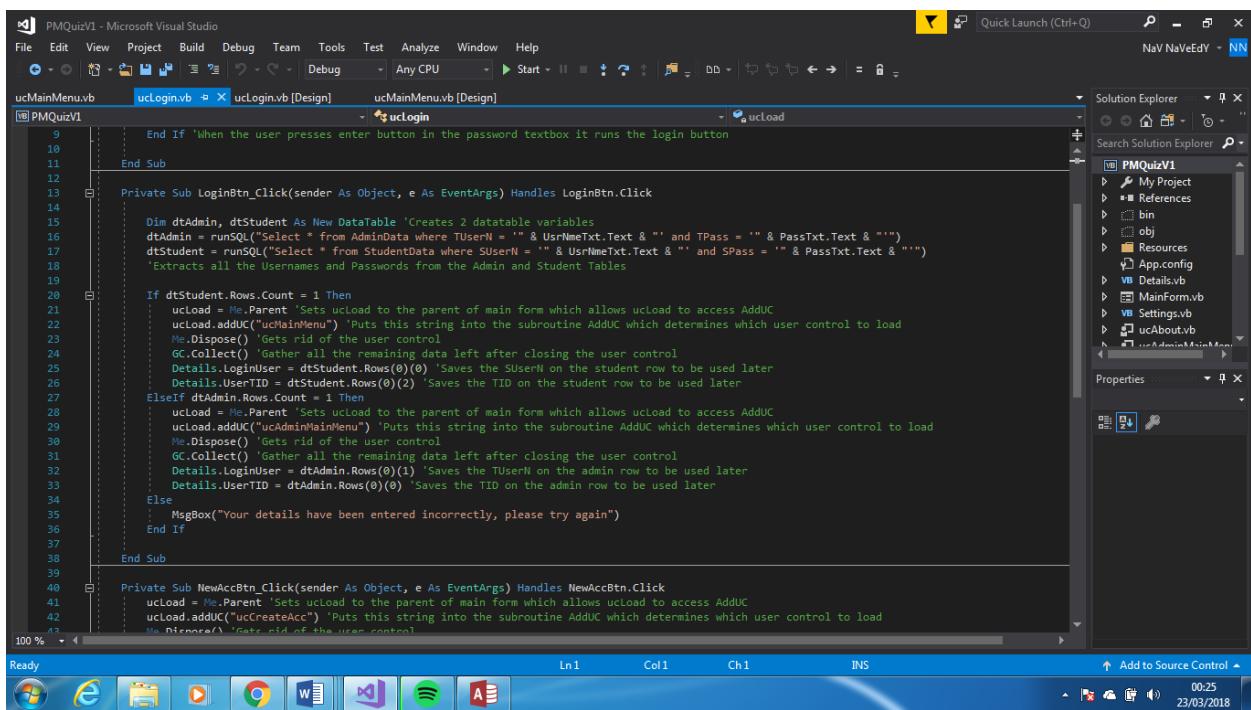
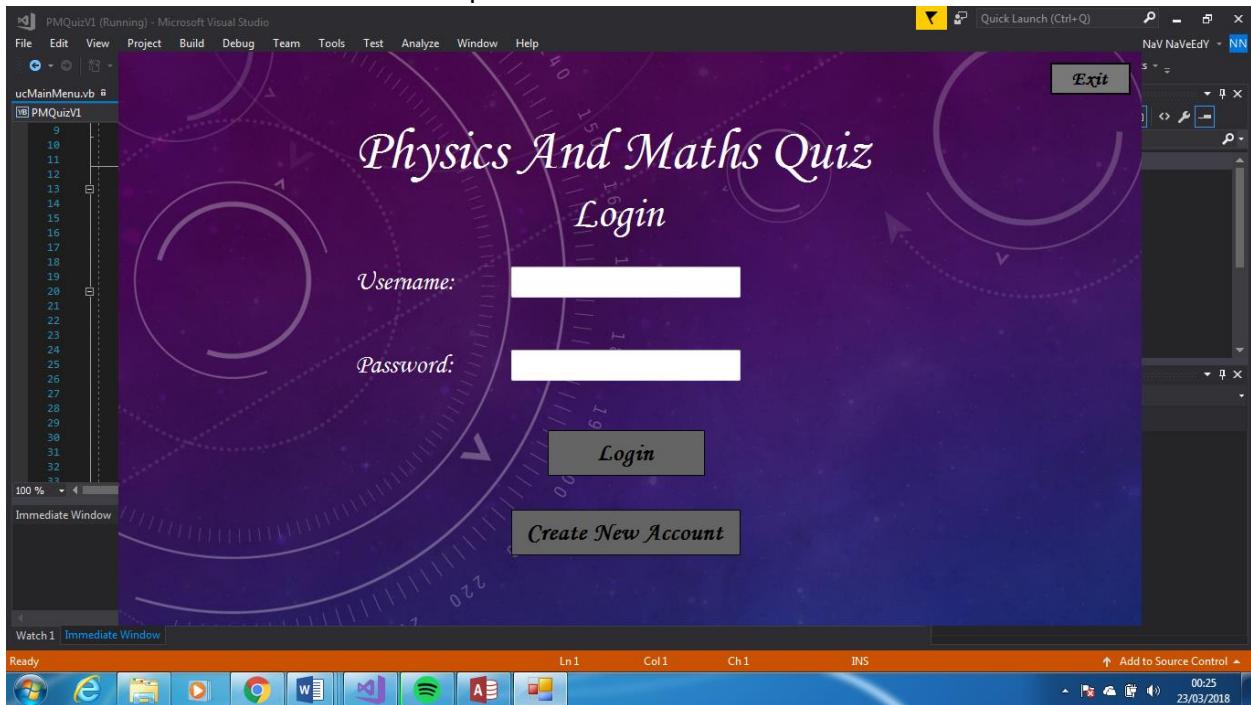
Test 6



Test 7

Candidate Name: Naveed Ali Rafeeq

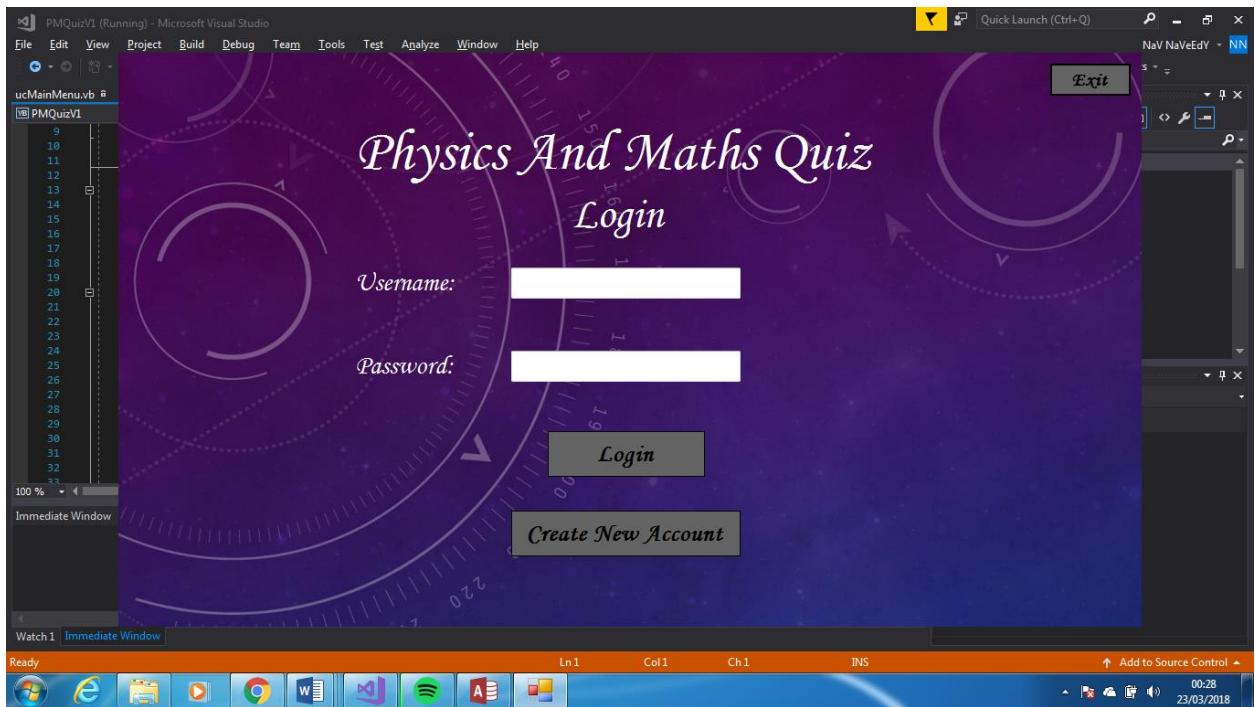
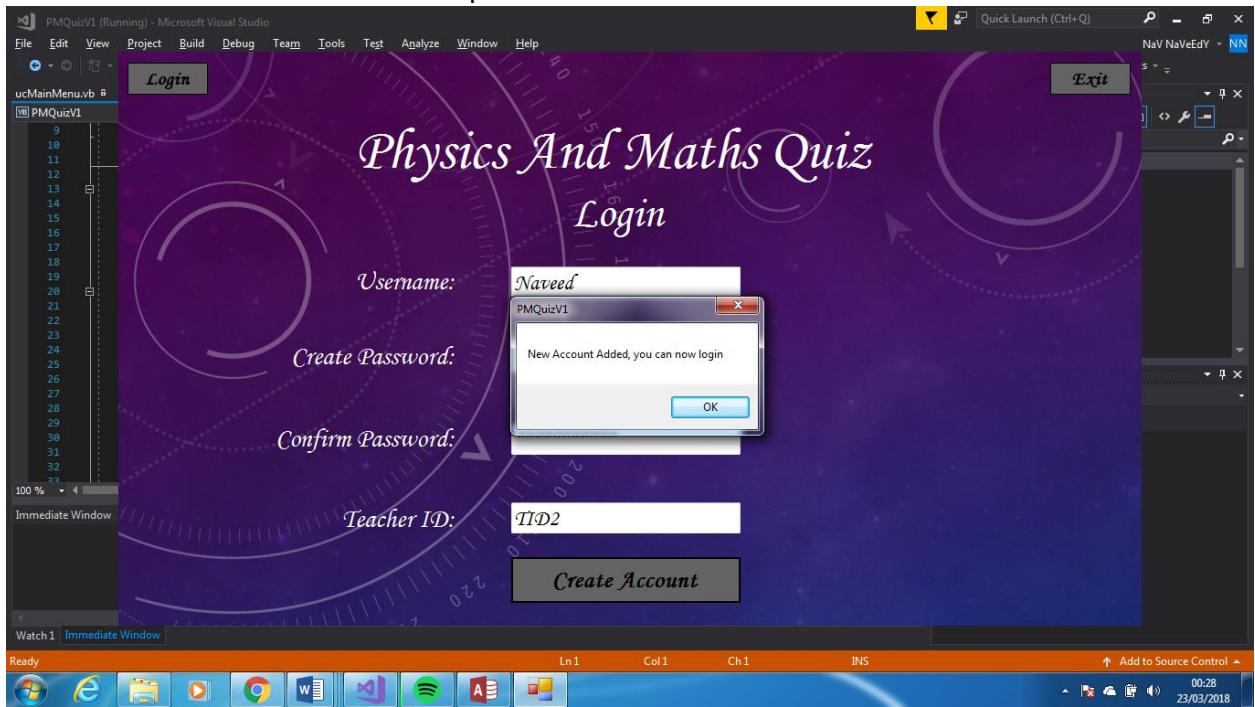
Candidate Number: 1904



## Test 8

Candidate Name: Naveed Ali Rafeeq

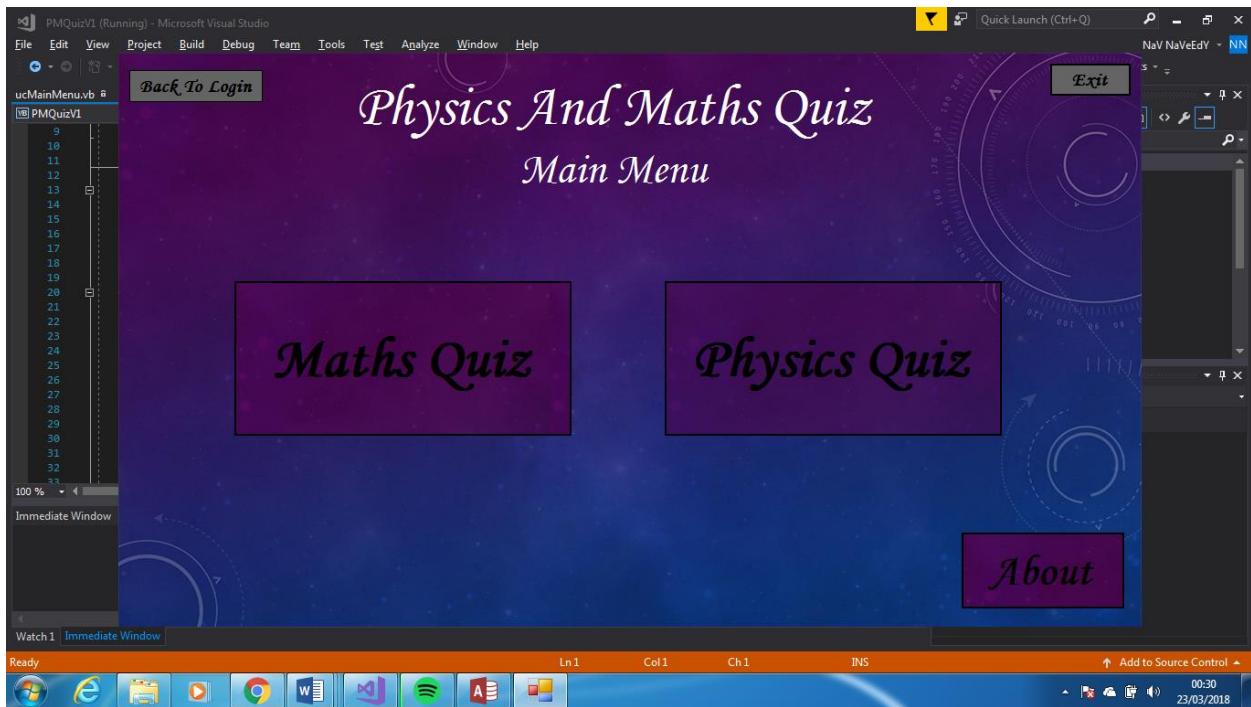
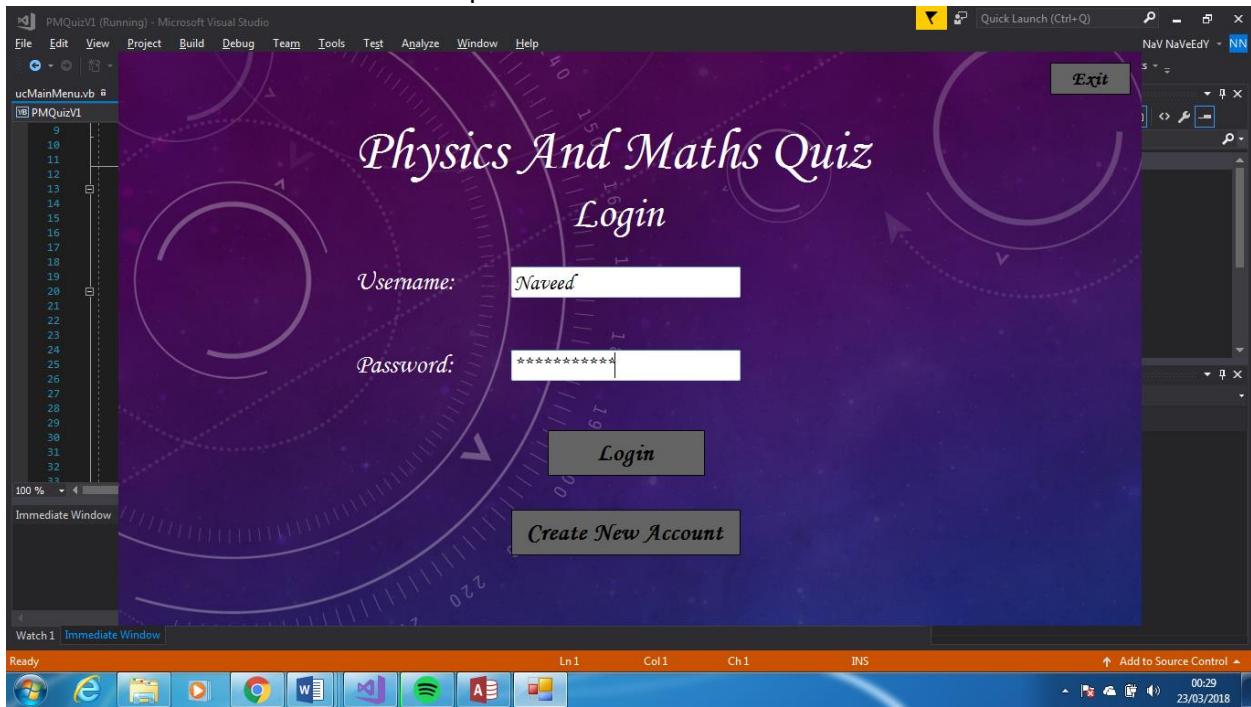
Candidate Number: 1904



## Test 9

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



## Interview with stakeholder – Prototype 1

I conducted an interview with one of my stakeholders Brendan (the Physics teacher) and asked for his opinions on my quiz. Even though, I have not started the question part of the program and have only done the login part I could still try and implement his feedback into my quiz for the next prototype.

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

Me: So, what do you think of the visual design of the project?

Brendan: I think that it is quite appealing especially to the younger audience; it brings a quite scientific feeling to it due to its background.

Me: Is there anything that you would change / add?

Brendan: Um, to be honest I'm not the biggest fan of the grey buttons that you have used, if you could maybe add buttons that suit the background more. For small buttons I guess it is fine but for much larger buttons I suggest making them the same / similar to the background so that they are not so out of place.

Me: What about the navigation in the program? Do you know where everything is and how to get to all the different pages?

Brendan: Yes, I love the fact that the pages have a quite simple design to them. It makes the quiz user friendly; keep that up for the rest of the program.

Me: So you do not have any problems with the navigation?

Brendan: Well one tiny thing you could improve is the Exit button.

Me: What about the Exit button?

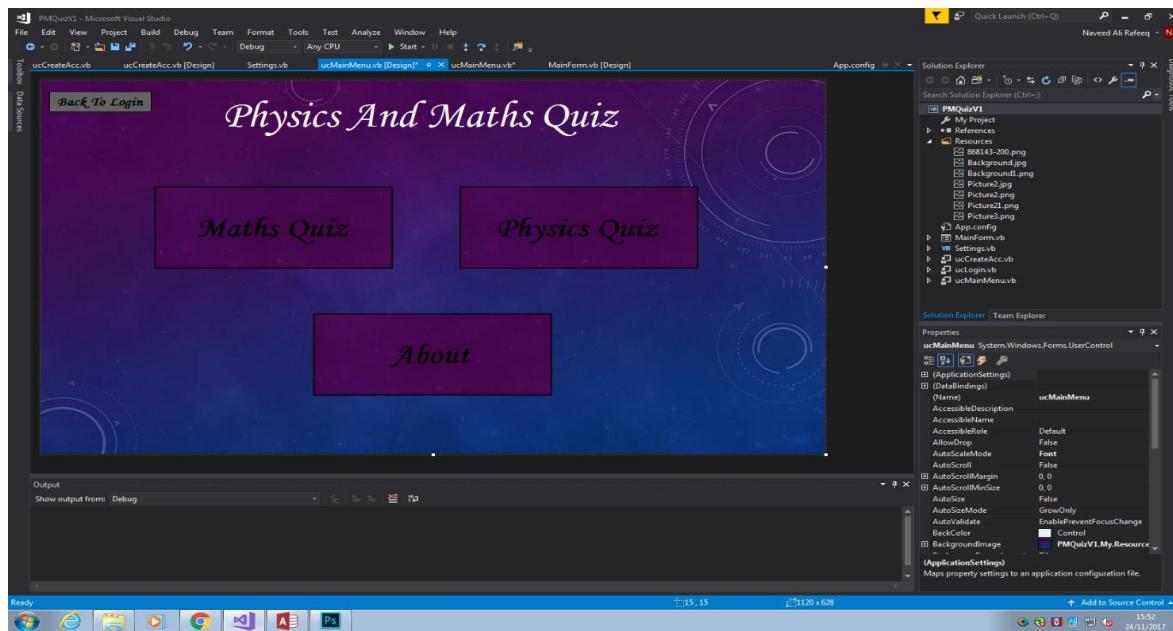
Brendan: Try to add some verification to it due to the fact that when I was using your quiz I accidentally clicked on the Exit button and the whole program closed. If I was taking a quiz and that happened I would be very frustrated, add a second option when you click it to make sure that they actually want to close the program.

Me: Alright thanks sir, I'll definitely implement these changes to my program.

Brendan: No problem.

## **Prototype 2**

### **User Control – Main Menu**



This User Control is the Main Menu for the students; it is used to navigate between the quizzes. It consists of 4 buttons that will either take you back to the login UC, the About UC, the Maths Quiz UC or the Physics Quiz UC. I added a different background to this UC just so there is a change however; it is the same theme as the other background so matches quite well.

As Brendan suggested I changed the background of my buttons from dark grey to the same as the background, which to be fair looks much better than big grey plain buttons. I used the properties menu of the buttons to select the same image as the background from my resources. These new buttons also caused some problems as well, due to the fact that it was a custom image and not one already given, the buttons would not highlight/become opaque when the mouse hovered over them. This was annoying due to the fact that the other buttons would become highlighted, so it seemed like there was no button there.

Also, since the user control Main Menu started to lag when I loaded it since I had full background sized images to load for each



button therefore I cropped the original background image into a smaller one fit for the size of the buttons.



I then used Photoshop to make the picture opaque, this therefore looks like a highlighted version of the background of the button.

```

Private Sub MQuizbtn_MouseEnter(sender As Object, e As EventArgs) Handles MQuizbtn.MouseEnter
    MQuizbtn.BackgroundImage = My.Resources.BgBlurSmall 'Makes the Maths Quiz background image the same as the resource image
    'This sub loads a different image when the mouse hovers over the Maths Quiz button
End Sub

Private Sub PQuizbtn_MouseEnter(sender As Object, e As EventArgs) Handles PQuizbtn.MouseEnter
    PQuizbtn.BackgroundImage = My.Resources.BgBlurSmall 'Same as the above but with the Physics Quiz button
End Sub

Private Sub Aboutbtn_MouseEnter(sender As Object, e As EventArgs) Handles Aboutbtn.MouseEnter
    Aboutbtn.BackgroundImage = My.Resources.BgBlurSmall ' Same as the above two but with the About button
End Sub

```

These 3 subs were then implemented in order to load the opaque picture when the mouse would hover over the button. So, I managed to fix my button highlight problem as well as my sort out my lag issue for the main menu User Control.

## Editing The Exit Button

```

Public Sub CloseBtn_Click(sender As Object, e As EventArgs) Handles CloseBtn.Click
    'This IF statement asks the user whether they are sure if they want to leave
    If MsgBox("Are you sure you want to exit the program? ", MsgBoxStyle.YesNo, "Exit") = MsgBoxResult.Yes Then
        End 'This button closes the whole program
    End If
End Sub
End Class

```

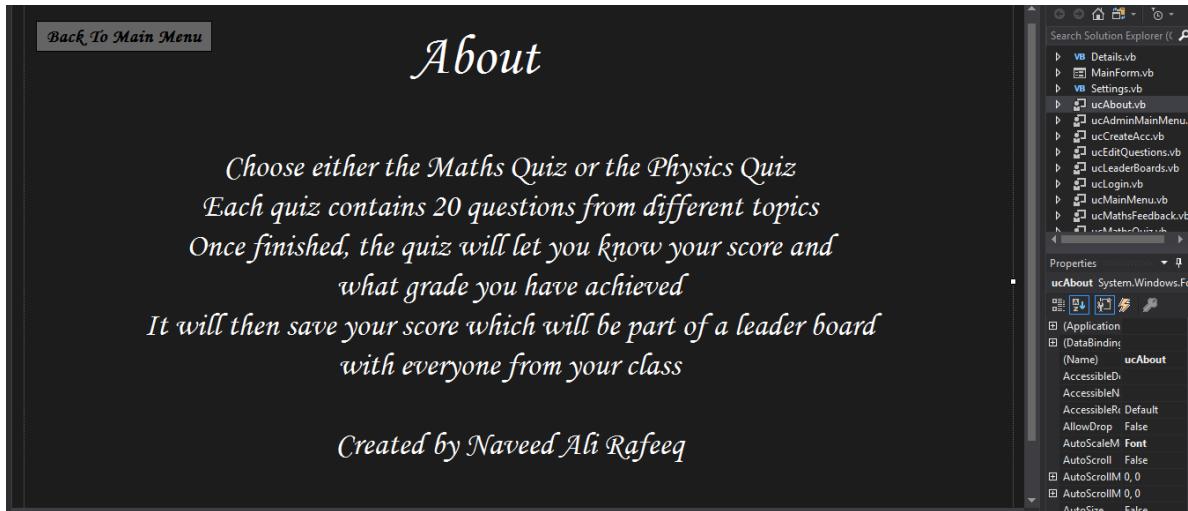
Due to my interview with Brendan I changed the Exit button so that now a message box appears asking the user if they are sure that they want to exit the program and only if they click

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

yes then the program closes. Therefore, if you accidentally click it then a form of verification will appear.

## User Control – About



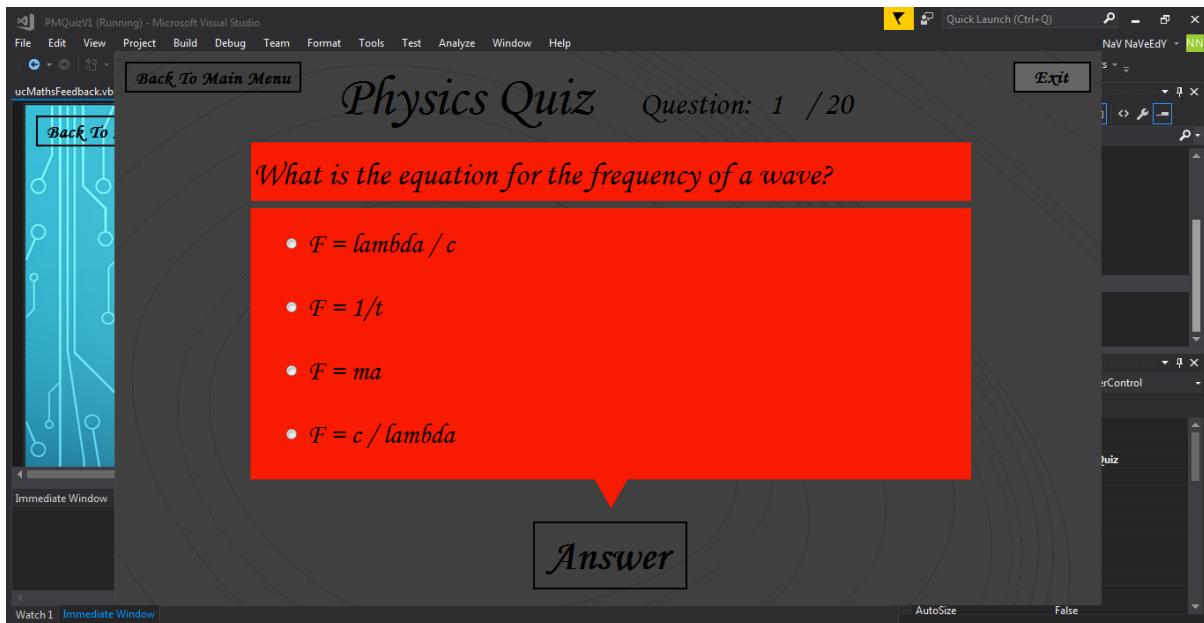
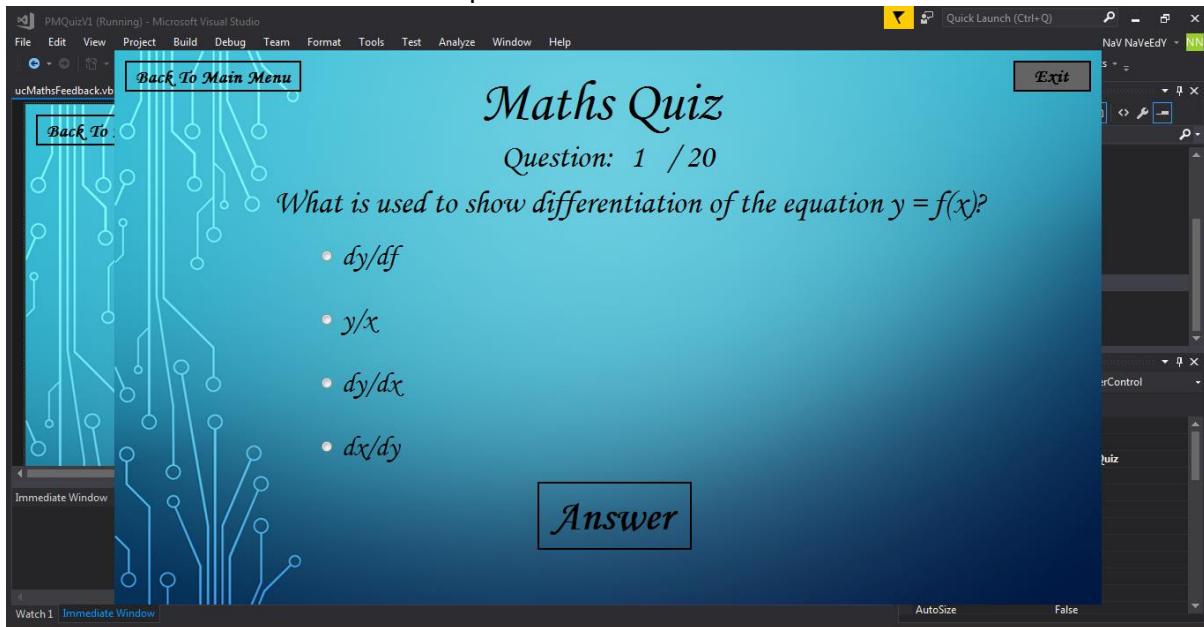
This was by far my simplest User Control since it is just a label with text informing the user how the quiz works as well as a button to take you back to the main menu.

```
Public Class ucAbout
    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
        Dim ucLoad As MainForm
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    End Sub
End Class
```

## User Control – Maths Quiz & Physics Quiz

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



These two user controls are the main components of the program these are the actual quiz parts, each work in the same way but with different backgrounds. The way the quiz works is every time the UC loads a random question and four answers which are ordered randomly. The user chooses an answer by clicking the corresponding radio button and then the answer button, if its correct a correct message box appears otherwise an incorrect message box appears. Then when you close the message box the UC refreshes and therefore a new question and answers.

```

Private Sub ucMathsQuiz_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    Dim dtLdQtn As DataTable 'Assigns the datatable to a variable
    Dim RowAmount, ranNum As Integer 'Creates 2 integer variables

    dtLdQtn = runSQL("Select MQuestion from MQuestions") 'Loads the MQuestion field from the MQuestions table in the access database
    RowAmount = dtLdQtn.Rows.Count 'Counts the amount of rows in the table and assigns the integer to RowAmount
    ranNum = CInt(Rnd() * (RowAmount - 1)) + 1 'This generates a random number from how many rows there was
    If ranNum = RowAmount Then
        ranNum = 0
    End If 'If statement to make sure that the first row in the datatable is included in the random selection
    QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random question from the table

End Sub
End Class

```

I started developing this sub that runs when the Maths Quiz Loads, it loads the all the questions from the MQuestions table into dtLdQtn. It then counts the number of rows in the Data Table and assigns it to the variable RowAmount. The ranNum variable then stores a random number in the RowAmount. If the ranNum is equal to the number of rows in the Data Table then it makes the ranNum equal to 0, the reason for this was due to the fact that when listing the row number using co-ordinates the first row is equal to 0. Therefore, if the co-ordinate for the row was equal to the row amount then it would not be able to find that row.

However, I realised 2 things about my sub so far. Firstly, the code is redundant since the If statement is not needed as well as the + 1 due to the – 1 would account for the co-ordinate problem. Secondly, even though the code would run a random number from the Table in the database it would always choose the same random number. Therefore, not making it random so I changed it to the following:

```

Private Sub ucMathsQuiz_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    Dim dtLdQtn As DataTable 'Assigns the datatable to a variable
    Dim RowAmount, ranNum As Integer 'Creates 2 integer variables

    dtLdQtn = runSQL("Select * from MQuestions") 'Loads the MQuestion field from the MQuestions table in the access database
    RowAmount = dtLdQtn.Rows.Count 'Counts the amount of rows in the table and assigns the integer to RowAmount
    Randomize()
    ranNum = Rnd() * (RowAmount - 1) 'This generates a random number from how many rows there was
    QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random question from the table

```

The randomize sub is part of the ‘VB.Math’ Library and this makes the Rnd function random each time, hence it becomes a random number

generat

or.

```

Private Function RanFunction() As Integer

    Dim RNum As Integer
    Randomize()
    RNum = Rnd() * (3) + 1
    Return RNum

End Function

```

```

QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random question from the table

Dim counter, RanArray(3) As Integer
Dim Repeat As Boolean = True
counter = 0

While Repeat = True
    For Each x In RanArray
        If RanFunction() = RanArray(x) Then
            Repeat = True
        Else
            RanArray(counter) = RanFunction()
            counter = counter + 1
        End If
    Next
End While

Rbtn1.Text = dtLdQtn.Rows(ranNum)(RanArray(0))
Rbtn2.Text = dtLdQtn.Rows(ranNum)(RanArray(1))
Rbtn3.Text = dtLdQtn.Rows(ranNum)(RanArray(2))
Rbtn4.Text = dtLdQtn.Rows(ranNum)(RanArray(3))
'Loads all the answers to the corresponding buttons

```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

Next, I created 3 variables; a counter, an array (containing 4 values) and a Boolean value. I created a loop that loops while the Boolean variable 'repeat' is true. Then it calls upon the function I created that generates a random number between 1 – 4, after it checks if this value is already in the array. If it is then it repeats the loop again otherwise it assigns the value to the first part in the array and then increments the counter, so the next value will be placed into the second part of the array and so on. Therefore, when I loaded all the array items into the text of each radio button I hoped for no repetition and a random order. Unfortunately, this algorithm did not work, it generated random answers on the same row for the question however it kept outputting identical answers. So, I decided to scratch this algorithm and worked on a new one.

After a while I managed to recall a problem we had to solve back in year 12 that asked us to shuffle a pack of cards. The way we managed to do this was by having an array containing 52 items with each item being a different card. Then, the algorithm would select a random item in the array and copy it into a temp value that was separate to the array. Next, the algorithm would select another random item in the array and copy its value into the location of where the first value was selected. Finally, the second items value would now equal to the temp value. This process was then placed in a loop and then repeated multiple times in order to shuffle the pack and essentially now making a random order. I decided implemented this algorithm into mine.

```
QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random question from the table

Dim ARanArray() As Integer = {1, 2, 3, 4} 'Creates an Array full of integers between 1 and 4
Dim tempAns As String 'Creates the temporary string variable
Dim ARnd1, ARnd2 As Integer 'Creates 2 integer variables

For x = 0 To 9 'Loops 10 times

    Randomize()
    ARnd1 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assigns it to the first integer variable
    Randomize()
    ARnd2 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assagns it to the second integer variable

    tempAns = ARanArray(ARnd1) 'The string variable now contains the a part of the array thats a value of rnd1 along
    ARanArray(ARnd1) = ARanArray(ARnd2) 'The value that was contained in that part of the array is now equal to the value
    'that was in the array rnd2 along
    ARanArray(ARnd2) = tempAns 'Now the value of the original rn1 is now placed into the array rnd2 along
    'It has now swapped the two numbers around within the array
Next
'This loop therefore swaps different numbers 10 times essentially a shuffling the array

Rbtn1.Text = dtLdQtn.Rows(ranNum)(ARanArray(0))
RBtn2.Text = dtLdQtn.Rows(ranNum)(ARanArray(1))
RBtn3.Text = dtLdQtn.Rows(ranNum)(ARanArray(2))
RBtn4.Text = dtLdQtn.Rows(ranNum)(ARanArray(3))
'Loads all the answers within the array and equals them to the radio button text's
Details.QtnTotal += 1 'Increments the total question integer
QNumLbl.Text = CStr(Details.QtnTotal) 'The question number is equal to the question total but has been converted into string form
FeedBackLd() 'Runs the sub routine every time the User Control Loads

End Sub
```

All of the above is similar to the algorithm I mentioned before with the 2 choices for the random array items to be swapped are the variables ARnd1 and ARnd2. I looped it 10 times because this seemed like it would be enough swaps to randomise the array. Then, the values within the array are used to select the co-ordinates of which column to output from column 1

to 4. (The 0 column is not included since this is the question within the question tables in the database).

```

1 - Microsoft Visual Studio
View Project Build Debug Team Tools Test Analyze Window Help
ucPhysicsQuiz.vb* ucPhysicsQuiz.vb [Design]* ucMathsQuiz.vb* ucMathsQuiz.vb [Design]*
Details LoginUser
Public Class Details 'These variables are now global therefore, when the uc's is disposed the variable data is not taken with it
    Public Shared QtnRight, QtnTotal As Integer

```

Next I created these 2 global variables, the reason for this was due to the fact that the way my quiz works is by the User Control refreshing every time the answer button is clicked. This meant that I had to make these variables global since my code would dispose all the data when a User Control is cleared, therefore it kept deleting the variables that would keep score of the question number as well as the users score.

```

Public Sub AnsBtn_Click(sender As Object, e As EventArgs) Handles AnsBtn.Click

    Dim dtLdAns As New DataTable

    If Rbtn1.Checked Then 'If the 1st radio button is ticked then
        dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & Rbtn1.Text & "'") 'Loads from the database the rows
        'which have the MA1 value equal to the radiobutton text
        If dtLdAns.Rows.Count > 0 Then 'If there is more than 0 rows left then it outputs a message saying they are correct
            MsgBox("                CORRECT!")

            Onto The Next Question  ")
            Details.QtnRight += 1 'If this message box is displayed then the question right variable increments
        Else
            MsgBox("                INCORRECT!"

            Onto The Next Question  ")
        End If
    End If

    If RBtn2.Checked Then 'Same but with the 2nd radio button
        dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & RBtn2.Text & "'")
        If dtLdAns.Rows.Count > 0 Then
            MsgBox("                CORRECT!")

            Onto The Next Question  ")
            Details.QtnRight += 1
        Else
            MsgBox("                INCORRECT!"

            Onto The Next Question  ")
        End If
    End If

```

```

If RBtn3.Checked Then 'Same but with the 3rd radio button
    dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & RBtn3.Text & "'")
    If dtLdAns.Rows.Count > 0 Then
        MsgBox("                               CORRECT!")
        Onto The Next Question  ")
        Details.QtnRight += 1
    Else
        MsgBox("                               INCORRECT!")
        Onto The Next Question  ")
    End If
End If

If RBtn4.Checked Then 'Same but with the 4th radio button
    dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & RBtn4.Text & "'")
    If dtLdAns.Rows.Count > 0 Then
        MsgBox("                               CORRECT!")
        Onto The Next Question  ")
        Details.QtnRight += 1
    Else
        MsgBox("                               INCORRECT!")
        Onto The Next Question  ")
    End If
End If

ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
ucLoad.addUC("ucMathsQuiz") 'Puts this string into the subroutine AddUC which determines which user control to load
Me.Dispose() 'Gets rid of the user control
GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

```

This sub runs when the answer button is clicked, it mainly consists of 4 If statements that all have another If statement within them, each for the 4 radio buttons. So, what the statement does is if the radio button is checked when the answer button is pressed it loads all of the MQuestions table into the Data Table and then loads the row that is equal to the text of that radio button. If there is more than one row left (essentially if its in the table in that field), a message box is displayed telling the user they are correct and thus incrementing their score by 1. If there are 0 rows left in the Data Table, then it displays a message box telling the user they are incorrect. The sub then refreshes the User Control so that a new question and answers can now load.

```

Private Sub FeedBackLd()
    If Details.QtnTotal = 4 Then
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucMathsFeedback") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    End If
End Sub

```

This sub was created so that the program would not be stuck in an infinite loop of questions. What it does is it loads the next User Control of the quiz, which is the feedback section, when the total question number is equal to whatever you decide. Since I was still testing my Quiz I put the value to 4 so that I only had to answer 3 questions and then I could move onto the feedback.

```
'Loads all the answers within the array and equals them to the radio button text's
Details.QtnTotal += 1 'Increments the total question integer
QNumLbl.Text = CStr(Details.QtnTotal) 'The question number is equal to the question total but has been converted into string form
FeedBackLd() 'Runs the sub routine every time the User Control Loads

End Sub
```

This code is placed at the end of the sub that runs when the Quiz UC loads the Question total variable will be incremented, then the label at the top would equal to the variable (after being converted into string). This meant that the first question would not read 0 but 1 since the first time it is loading the Quiz UC the question total variable will get incremented straight away. Lastly, it runs the FeedBackLd sub so that it can check whether the Question total variable is equal to it, so it can change to the feedback UC. Later on, I change it to 21 so that 20 questions will be outputted before we go onto the Feed Back page.

```
Public Class ucMathsQuiz

    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

        If MsgBox("Are you sure you want to leave, your quiz score won't be saved?", MsgBoxStyle.YesNo, "Maths Quiz") = MsgBoxResult.Yes Then

            ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
            ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
            Me.Dispose() 'Gets rid of the user control
            GC.Collect() 'Gather all the remaining data left after closing the user control

        End If
    End Sub
```

The final part of this User Control is the Back Button, which takes you back to the Main Menu, however as requested by one of my stakeholders Brendan Foster, I added some validation into it just in case they accidentally clicked the button or if they suddenly change their mind then they will not lose all their progress made in the quiz. When the Back To Main Menu button is clicked a message box appears that asks them whether they are sure they want to leave. If they click yes then it loads the Main Menu UC, if they click no then it just closes the text box and they can carry on with their quiz.

```
Private Sub ucMainMenu_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Details.QtnTotal = 0 'This resets the variable question total back to 0
    Details.QtnRight = 0 'This resets the variable question right back to 0
End Sub
```

Finally, I added these 2 lines of code into a sub that loads when the Main Menu User control loads. This means that the user can do many quizzes without exiting the program because to access a quiz you have to be on the main menu and therefore the 2 global variable will be reset before loading a new quiz.

**\*The whole code for the Physics Quiz is almost identical to the Maths Quiz apart from the fact that it loads questions and answers from the Physics Table instead of the Maths Table from the database.**

```

Public Class ucPhysicsQuiz
    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
        If MsgBox("Are you sure you want to leave, your quiz score won't be saved? ", MsgBoxStyle.YesNo, "Physics Quiz") = MsgBoxResult.Yes Then
            ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
            ucLoad.AddUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
            Me.Dispose() 'Gets rid of the user control
            GC.Collect() 'Gather all the remaining data left after closing the user control
        End If
    End Sub

    Private Sub ucPhysicsQuiz_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim dtLdQtn As DataTable 'Assigns the datatable to a variable
        Dim RowAmount, ranNum As Integer 'Creates 2 integer variables

        dtLdQtn = runSQL("Select * from PQuestions") 'Loads the PQuestion field from the PQuestions table in the access database
        RowAmount = dtLdQtn.Rows.Count 'Counts the amount of rows in the table and assigns the integer to RowAmount
        Randomize()
        ranNum = Rnd() * (RowAmount - 1) 'This generates a random number from how many rows there was
        QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random question from the table

        Dim ARanArray() As Integer = {1, 2, 3, 4} 'Creates an Array full of integers between 1 and 4
        Dim tempAns As String 'Creates the temporary string variable
        Dim ARnd1, ARnd2 As Integer 'Creates 2 integer variables
    
```

```

For x = 0 To 9 'Loops 10 times

    Randomize()
    ARnd1 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assigns it to the first integer variable
    Randomize()
    ARnd2 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assigns it to the second integer variable

    tempAns = ARanArray(ARnd1) 'The string variable now contains the a part of the array thats a value of rnd1 along
    ARanArray(ARnd1) = ARanArray(ARnd2) 'The value that was contained in that part of the array is now equal to the value
    'that was in the array rnd2 along
    ARanArray(ARnd2) = tempAns 'Now the value of the original rnd1 is now placed into the array rnd2 along
    'It has now swapped the two numbers around within the array
Next
'This loop therefore swaps different numbers 10 times essentially a shuffling the array

Rbtn1.Text = dtLdQtn.Rows(ranNum)(ARanArray(0))
Rbtn2.Text = dtLdQtn.Rows(ranNum)(ARanArray(1))
Rbtn3.Text = dtLdQtn.Rows(ranNum)(ARanArray(2))
Rbtn4.Text = dtLdQtn.Rows(ranNum)(ARanArray(3))
'Loads all the answers within the array and equals them to the radio button text's

Details.QtnTotal += 1 'Increments the total question integer
QNumLbl.Text = CStr(Details.QtnTotal) 'The question number is equal to the question total but has been converted into string form
FeedBackId() 'Runs the sub routine every time the User Control Loads
End Sub

```

```

Private Sub AnsBtn_Click(sender As Object, e As EventArgs) Handles AnsBtn.Click
    Dim dtLdAns As New DataTable

    If Rbtn1.Checked Then 'If the 1st radio button is ticked then
        dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & Rbtn1.Text & "'") 'Loads from the database the rows
        'which have the PA1 value equal to the radiobutton text
        If dtLdAns.Rows.Count > 0 Then 'If there is more than 0 rows left then it outputs a message saying they are correct
            MsgBox("                CORRECT!")

            Onto The Next Question ")
            Details.QtnRight += 1 'If this message box is displayed then the question right variable increments
        Else
            MsgBox("                INCORRECT!"

            Onto The Next Question ")
        End If
    End If

    If RBtn2.Checked Then 'Same but with the 2nd radio button
        dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & RBtn2.Text & "'")
        If dtLdAns.Rows.Count > 0 Then
            MsgBox("                CORRECT!")

            Onto The Next Question ")
            Details.QtnRight += 1
        Else
            MsgBox("                INCORRECT!"

            Onto The Next Question ")
        End If
    End If

```

```

    If RBtn3.Checked Then 'Same but with the 3rd radio button
        dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & RBtn3.Text & "'")
        If dtLdAns.Rows.Count > 0 Then
            MsgBox("                CORRECT!")

            Onto The Next Question ")
            Details.QtnRight += 1
        Else
            MsgBox("                INCORRECT!"

            Onto The Next Question ")
        End If
    End If

    If RBtn4.Checked Then 'Same but with the 4th radio button
        dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & RBtn4.Text & "'")
        If dtLdAns.Rows.Count > 0 Then
            MsgBox("                CORRECT!")

            Onto The Next Question ")
            Details.QtnRight += 1
        Else
            MsgBox("                INCORRECT!"

            Onto The Next Question ")
        End If
    End If

```

```

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.addUC("ucPhysicsQuiz") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

Private Sub FeedBackLd()
If Details.QntTotal = 4 Then
    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.addUC("ucPhysicsFeedback") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control
End If
End Sub
End Class

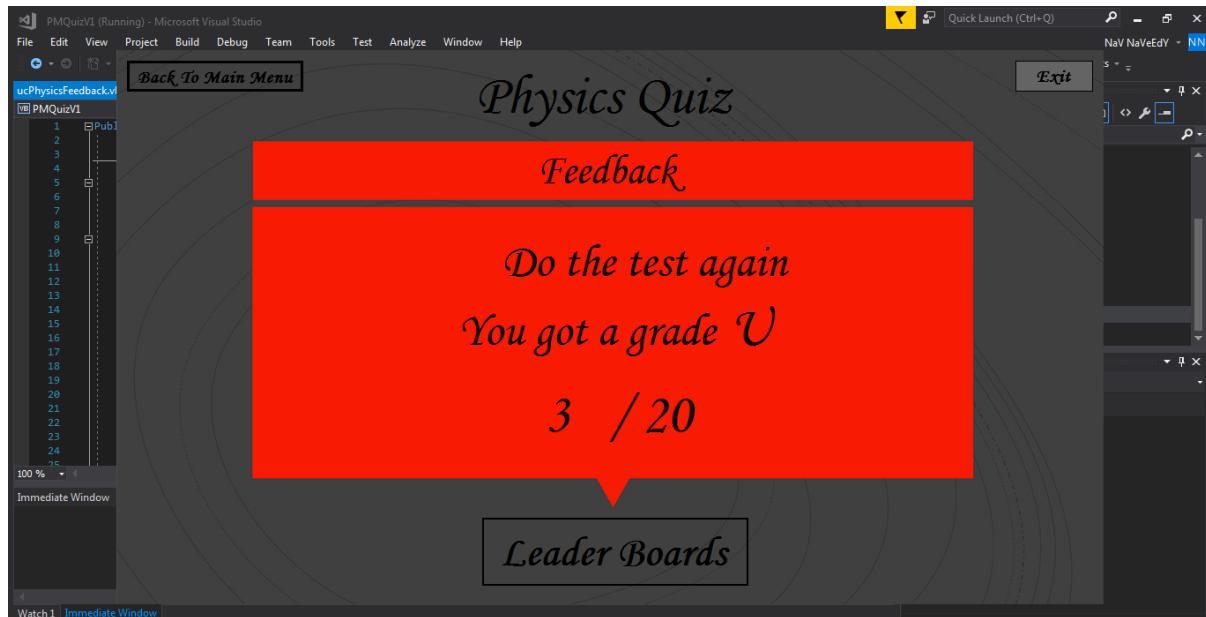
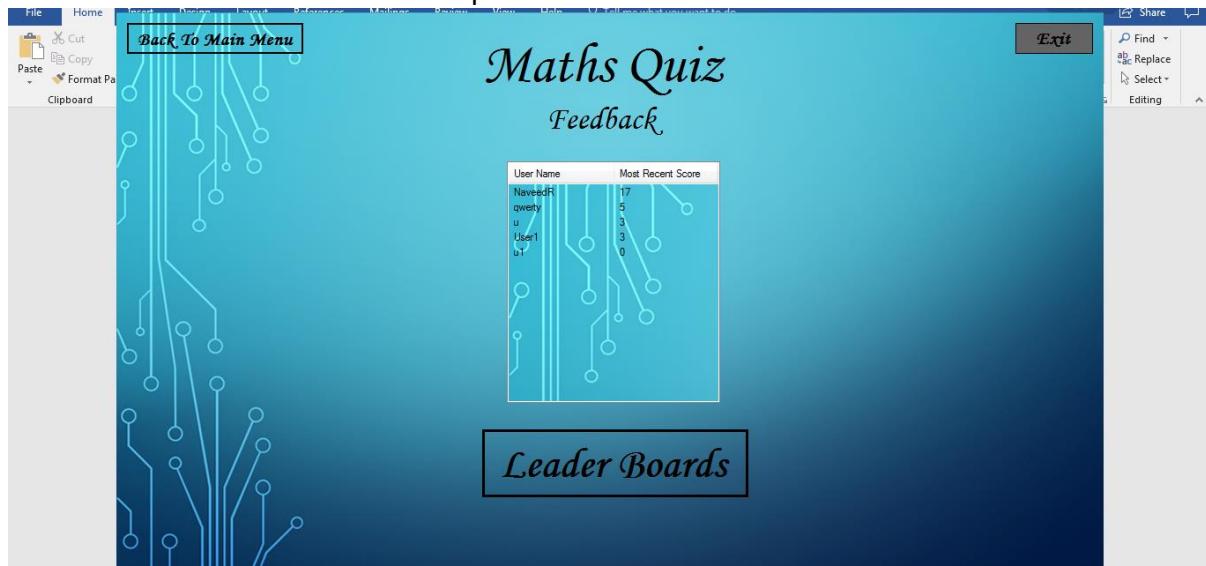
```

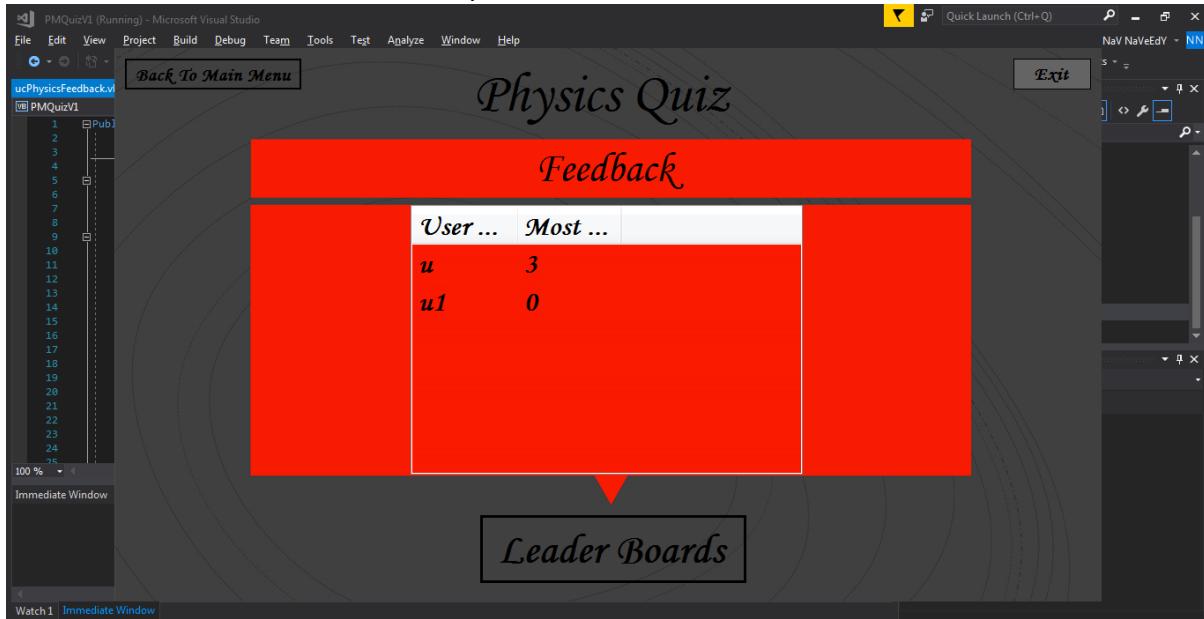
## User Control – Maths Feed Back & Physics Feed Back



Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904





These User Controls give the users feed back depending on what score they got within the quiz. I used many different labels in the same statement due to the fact that some needed to be changed according to the users score. Therefore, I used the following select statement:

```

Public Class ucMathsFeedback
    Dim ucLoad As MainForm

    Private Sub ucMathsFeedback_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        QtnRLbl.Text = Details.QtnRight 'The total questions the user got correct is assigned to the label

        Select Case Details.QtnRight 'The select case assigns a grade and a message to each value of QtnRight global variable
            Case Is > 17
                GrdLbl.Text = "A*"
                FBackLbl.Text = "You're A Genius"
            Case Is = 16 Or 17
                GrdLbl.Text = "A"
                FBackLbl.Text = "Pretty Good"
            Case Is = 14 Or 15
                GrdLbl.Text = "B"
                FBackLbl.Text = "Decent"
            Case Is = 12 Or 13
                GrdLbl.Text = "C"
                FBackLbl.Text = "You Passed"
            Case Is = 10 Or 11
                GrdLbl.Text = "D"
                FBackLbl.Text = "You Can Do Better"
            Case Is = 8 Or 9
                GrdLbl.Text = "E"
                FBackLbl.Text = "Unlucky"
            Case Is < 8
                GrdLbl.Text = "U"
                FBackLbl.Text = "Do the test again"
        End Select
    End Sub

```

So when the User Control loads it firstly assigns the Question Right score label to the global variable score, which is the number they got right out of 20 in the text. Then the select case statement is pretty self-explanatory, depending on what range their score lands in it will output a certain grade and message with it.

```

Public Class Details 'These variables are now global therefore, when the uc's is disposed the variable data is not taken with it
    Public Shared QtnRight, QtnTotal As Integer
    Public Shared LoginUser, UserTID As String
End Class

```

Next, I created 2 more global variables that therefore can be called upon anywhere within the program.

```

Private Sub LoginBtn_Click(sender As Object, e As EventArgs) Handles LoginBtn.Click
    Dim dtAdmin, dtStudent As New DataTable 'Creates 2 datatable variables
    dtAdmin = runSQL("Select * from AdminData where TUserN = '" & UsrNmeTxt.Text & "' and TPass = '" & PassTxt.Text & "'")
    dtStudent = runSQL("Select * from StudentData where SUserN = '" & UsrNmeTxt.Text & "' and SPass = '" & PassTxt.Text & "'")
    'Extracts all the Usernames and Passwords from the Admin and Student Tables

    If dtStudent.Rows.Count = 1 Then
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
        Details.LoginUser = dtStudent.Rows(0)(0) 'Saves the SUserN on the student row to be used later
        Details.UserTID = dtStudent.Rows(0)(2) 'Saves the TID on the student row to be used later
    End If

```

After I added the last 2 lines to my Login UC, this meant that when a user would log in their username and TID would be saved to the global variable.

```

Dim dtScore As New DataTable 'Copies a table within the database into vb to be edited and then updated

dtScore = runSQL("SELECT * from MResults where SUserN = '" & Details.LoginUser & "'")
'Loads the query, which is the row equal to the username used when logging in, into the Data Table
If dtScore.Rows.Count > 0 Then 'If there is atleast a row in the datatable
    runSQL("DELETE * FROM MResults where SUserN = '" & Details.LoginUser & "'")
    'Delete the row
    runSQL("INSERT into MResults (SUserN, Results, TID) values ('" & Details.LoginUser & "','" &
        Details.QtnRight & "','" & Details.UserTID & "')")
    'Add the new row with the same username, TID but the updated score
Else 'If there is no rows then
    runSQL("INSERT into MResults (SUserN, Results, TID) values ('" & Details.LoginUser & "','" &
        Details.QtnRight & "','" & Details.UserTID & "')")
    'Add the new row with the username, TID and score
End If

```

I could then implement this If statement into the Maths Quiz Feedback Load sub. What this does is it updates your score within the Maths Results table by deleting your original details and adding it back in. If they have not done the quiz before then it would just straight insert it into the Maths Results table.

```

Private Sub ScoreLoad() 'Sub that loads the items (Username and Physics score) into the List View

    Dim dt As New DataTable 'Copies a table within the database into vb to be edited and then updated
    dt = runSQL("SELECT * from MResults where TID = '" & Details.UserTID & "' ORDER BY Results DESC;")
    'Loads all the rows from the Physics Results table that are in a certain class into the Data Table
    'Orders them from highest to lowest, therefore making it a leader board

    For x = 0 To dt.Rows.Count - 1
        Dim items(1) As String '1 dimensional array with 2 values
        items(0) = dt.Rows(x)(0) 'Username
        items(1) = dt.Rows(x)(1) 'Score
        'Adds the username and score into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items array into the LineNew
        LstScore.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table

End Sub

```

I created this sub that would be copied and amended later in every single list view I used within the program. It consists of loading the Maths Result table into a Data Table, then a loop that performs a certain amount of times depending on how many rows there are in the table. In the loop it fills the items of the columns one with the usernames and the other with their score.

Now, within my design and analysis I planned on using a sort algorithm in order to sort out the scores from highest to lowest. However, I discovered that I could easily not have to use an algorithm to sort out my values at all. Instead at the end of my SQL statement I ordered the table by DESC (descending order), this meant that the Data Table is now ranked by the highest to lowest. So, when it is placed into the list view it is already in descending order, I therefore saved much time and code by doing this and essentially makes my program more efficient due to it.

```

Private Sub LeadBtn_Click(sender As Object, e As EventArgs) Handles LeadBtn.Click

    'When the Leader Boards button is clicked the sub hides all the feedback text and the Leader board
    'button, then makes the listview visible
    LstScore.Visible = True
    FBackLbl.Visible = False
    Label2.Visible = False
    GrdLbl.Visible = False
    QtnRLbl.Visible = False
    Label3.Visible = False

End Sub

```

This sub just hides all of the feedback text and the Leader Boards button when the Leader Boards button is clicked.

```

End Select

LstScore.Visible = False 'The list view is hidden

Dim dtScore As New DataTable 'Copies a table within the database into vb to be edited and then updated

```

I then added this line to do sub that runs when the UC loads so that the leader boards is originally hidden and then you click the button to display it.

```
Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.AddUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control
End Sub
```

And finally, the back to main menu button is added to go back to the main menu.

**\*As with the Quizzes the feedback User Controls are also almost identical.**

```
Public Class ucPhysicsFeedback
    Dim ucLoad As MainForm

    Private Sub ucPhysicsFeedback_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        QtnRLbl.Text = Details.QtnRight 'The total questions the user got correct is assigned to the label

        Select Case Details.QtnRight 'The select case assigns a grade and a message to each value of QtnRight global variable
            Case Is > 17
                GrdLbl.Text = "A*"
                FBackLbl.Text = "You're A Genius"
            Case Is = 16 Or 17
                GrdLbl.Text = "A"
                FBackLbl.Text = "Pretty Good"
            Case Is = 14 Or 15
                GrdLbl.Text = "B"
                FBackLbl.Text = "Decent"
            Case Is = 12 Or 13
                GrdLbl.Text = "C"
                FBackLbl.Text = "You Passed"
            Case Is = 10 Or 11
                GrdLbl.Text = "D"
                FBackLbl.Text = "Unlucky"
            Case Is = 8 Or 9
                GrdLbl.Text = "E"
                FBackLbl.Text = "Peak"
            Case Is < 8
                GrdLbl.Text = "U"
                FBackLbl.Text = "Do The Test Again"
        End Select
    End Sub
```

```

LstScore.Visible = False 'The list view is hidden

Dim dtScore As New DataTable 'Copies a table within the database into vb to be edited and then updated

dtScore = runSQL("SELECT * from PResults where SUserName = '" & Details.LoginUser & "'")
'Loads the query, which is the row equal to the username used when logging in, into the Data Table
If dtScore.Rows.Count > 0 Then 'If there is at least one row in the datatable
    runSQL("DELETE * FROM PResults where SUserName = '" & Details.LoginUser & "'")
    'Delete the row
    runSQL("INSERT into PResults (SUserName, Results, TID) values ('" & Details.LoginUser & "','" &
        Details.QtnRight & "','" & Details.UserTID & "')")
    'Add the new row with the same username, TID but the updated score
Else 'If there is no rows then
    runSQL("INSERT into PResults (SUserName, Results, TID) values ('" & Details.LoginUser & "','" &
        Details.QtnRight & "','" & Details.UserTID & "')")
    'Add the new row with the username, TID and score
End If

ScoreLoad() 'Places the items into the List View

End Sub

```

```

Private Sub ScoreLoad() 'Sub that loads the items (Username and Physics score) into the List View

Dim dt As New DataTable 'Copies a table within the database into vb to be edited and then updated
dt = runSQL("SELECT * from PResults where TID = '" & Details.UserTID & "' ORDER BY Results DESC;")
'Loads all the rows from the Physics Results table that are in a certain class into the Data Table
'Orders them from highest to lowest, therefore making it a leader board

For x = 0 To dt.Rows.Count - 1
    Dim items(1) As String '1 dimensional array with 2 values
    items(0) = dt.Rows(x)(0) 'Username
    items(1) = dt.Rows(x)(1) 'Score
    'Adds the username and score into the row

    Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items array into the LineNew
    LstScore.Items.Add(LineNew) 'Adds the new line into the List View
Next 'Loop repeats for each row of the table

End Sub

Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
Me.Dispose() 'Gets rid of the user control
GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

```

```

Private Sub LeadBtn_Click(sender As Object, e As EventArgs) Handles LeadBtn.Click

'When the Leader Boards button is clicked the sub hides all the feedback text and the Leader board
'button, then makes the listview visible
LstScore.Visible = True
FBackLbl.Visible = False
Label2.Visible = False
GrdLbl.Visible = False
QtnRLbl.Visible = False
Label3.Visible = False

End Sub

```

## Details Class

```
Public Class Details 'These variables are now global therefore, when the uc's is disposed the variable data is not taken with it
    Public Shared QtnRight, QtnTotal As Integer
    Public Shared LoginUser, UserID As String
End Class
```

I stored my global variables in this Public Class so that I could call them in any other Class like I did in the Login UC, Quiz UCs and Feedback UCs.

## Main Form Update

The select statement that handles navigation between User Controls has now updated to this at the end of this prototype.

```
'Sub routine which takes in user controls, and makes a choice on which uc to load
Public Sub addUC(ByVal ucChoice As String)
    Dim newUC As UserControl 'variable newUC that adds new user control
    newUC = New ucLogin 'Assigns a User Control to newUC so the variable isn't empty

    Select Case ucChoice
        Case Is = "ucLogin"
            newUC = New ucLogin 'Loads the user control ucLogin
        Case Is = "ucMainMenu"
            newUC = New ucMainMenu 'Loads the user control ucMainMenu
        Case Is = "ucCreateAcc"
            newUC = New CreateAccBtn 'Loads the user control ucCreateAcc
        Case Is = "ucAbout"
            newUC = New ucAbout 'Loads the user control ucAbout
        Case Is = "ucMathsQuiz"
            newUC = New ucMathsQuiz 'Loads the user control ucMathsQuiz
        Case Is = "ucPhysicsQuiz"
            newUC = New ucPhysicsQuiz 'Loads the user control ucPhysicsQuiz
        Case Is = "ucMathsFeedback"
            newUC = New ucMathsFeedback 'Loads the user control ucMathsFeedback
        Case Is = "ucPhysicsFeedback"
            newUC = New ucPhysicsFeedback 'Loads the user control ucPhysicsFeedback
    End Select
    Me.Controls.Add(newUC) 'Adds the desired User Control
End Sub
```

## Testing Prototype 2

### Navigation:

Test Data	Test Number	Type Of Test Data	Output
LMC on Exit button	15	Valid - Extreme	Message box appears verifying the user's decision

RMC on buttons	<b>NA</b>	Invalid	Nothing
LMC not on the buttons	<b>NA</b>	Invalid	Nothing
LMC on any 'Back' button	<b>1</b>	Valid	Disposes current UC and loads another
Answering 20 questions in one of the Quiz User Controls	<b>2</b>	Valid	Disposes the Quiz UC and loads its corresponding Feedback UC
Answering 10 questions in one of the Quiz User Controls	<b>3</b>	Invalid	The 11 <sup>th</sup> question loads

I did more navigation testing to ensure that the whole program would switch screens smoothly, I mainly tested the new added User Controls. I also tested the Exit button again because I amended it so that it adds validation on whether the user definitely wants to close the program. As requested by one of my stakeholders within the 1<sup>st</sup> prototype interview.

#### Main Menu:

Test Data	Test Number	Type Of Test Data	Output
LMC on Maths Quiz button	<b>4</b>	Valid	Loads the Maths Quiz UC
LMC on Physics Quiz button	<b>5</b>	Valid	Loads the Physics Quiz UC
LMC on About button	<b>6</b>	Valid	Loads the About UC
Click where no buttons	<b>NA</b>	Invalid	Nothing
RMC on Maths Quiz button	<b>NA</b>	Invalid	Nothing
RMC on Physics Quiz button	<b>NA</b>	Invalid	Nothing
RMC on How To Play / About button	<b>NA</b>	Invalid	Nothing

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

With the Main Menu there is only 4 buttons to test and with one being tested already in the navigation above only the 3 buttons needed to be tested. They all just load up different User Controls.

#### Quiz User Controls:

Test Data	Test Number	Type Of Test Data	Output
LMC on a single radio button	7	Valid	Radio button becomes checked
LMC on 2 radio buttons at 1 time	8	Invalid	The new radio button becomes checked instead
LMC Answer button	9	Valid	Message box appears and new question loads
RMC Answer button	NA	Invalid	Nothing
RMC any radio button	NA	Invalid	Nothing
Click the back to Main Menu button	10	Valid – Extreme	Message box with an option to verify the user's decision. If yes then the new UC is displayed

To ensure that the Quiz sections of my program were to work as intended I tried breaking it using a variety of Test Data's. The valid extreme part especially checks whether can click the button to leave the quiz but then due to validation they can return to the quiz number they were on already.

#### Feedback User Controls:

Test Data	Test Number	Type Of Test Data	Output
Score 3	11	Valid	Displays 3 / 20
Score 25	NA	Invalid – Erroneous	N/A
Score 20	12	Valid – Extreme	Displays 20 / 20
Achieve 16 or more marks and a grade A will be displayed	13	Valid	"You got an A"

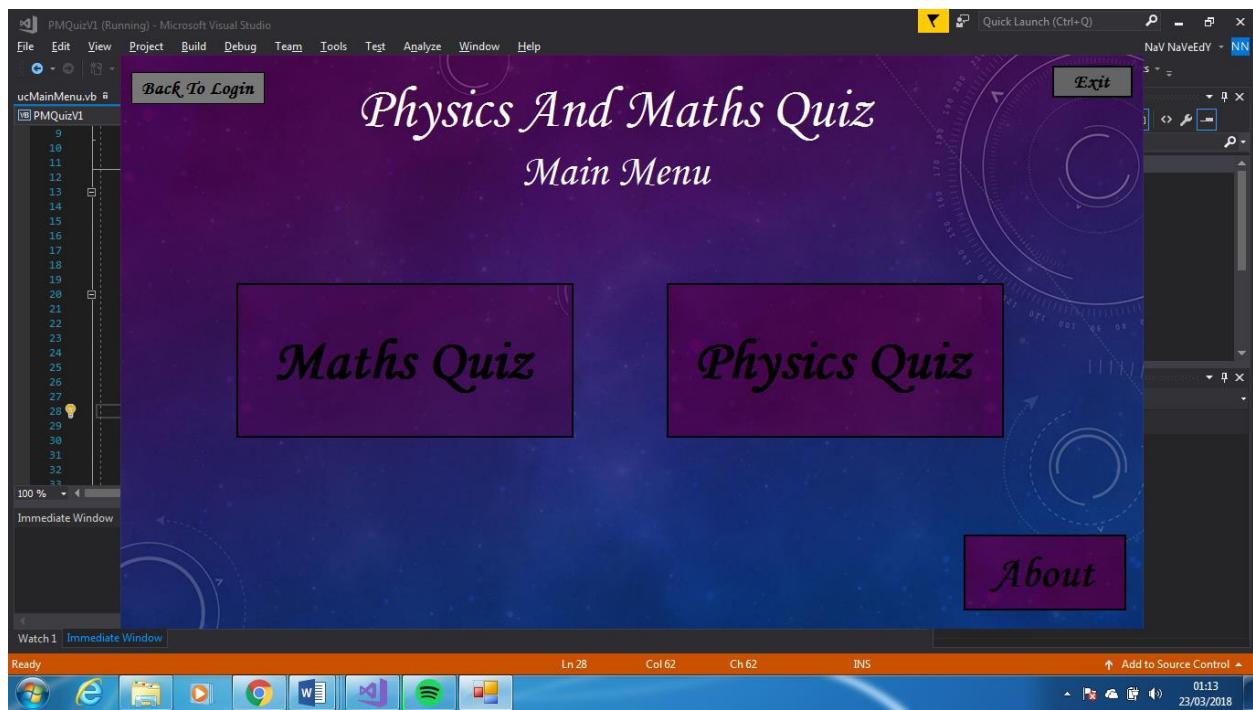
Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

Achieve a Z grade	<b>NA</b>	Invalid - Erroneous	<b>N/A</b>
Score -1	<b>NA</b>	Invalid	<b>N/A</b>
LMC on score	<b>NA</b>	Invalid – Erroneous	Nothing
Score 21	<b>NA</b>	Invalid – Extreme	<b>N/A</b>
LMC on Leader boards button	<b>14</b>	Valid	Loads a list view containing the usernames and scores

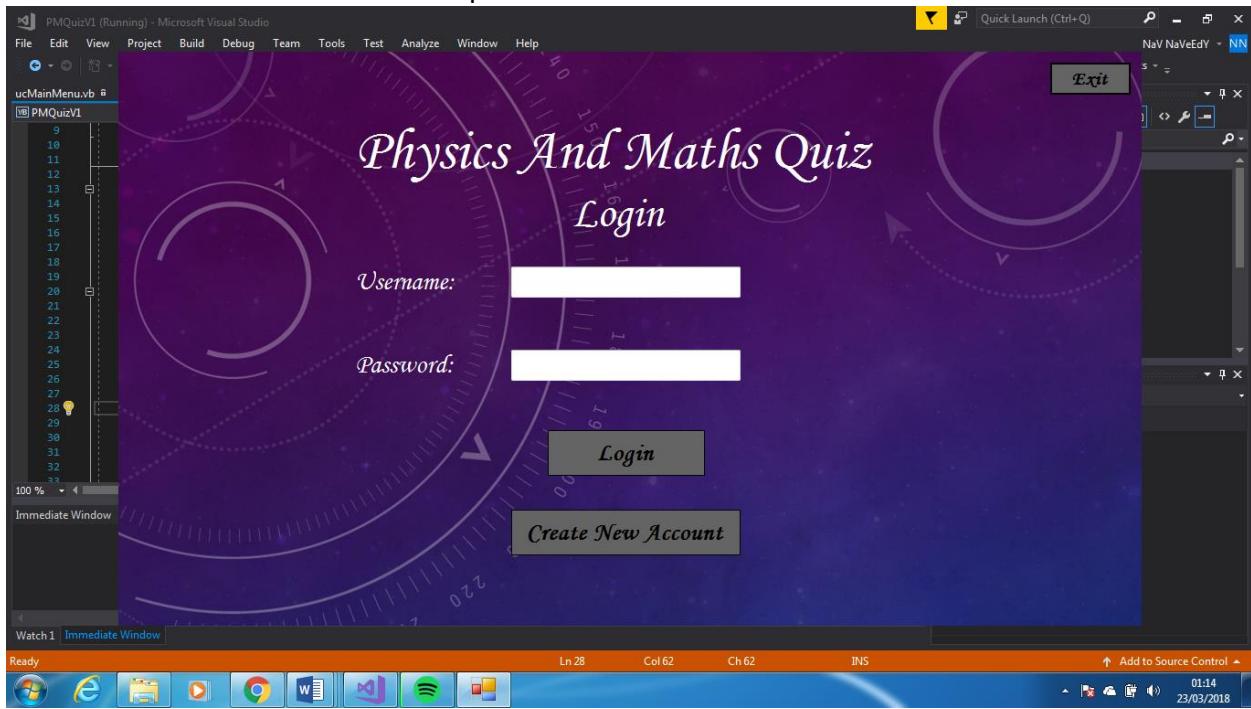
The purpose of these tests was to ensure that no invalid scores appear on the feedback user control.

### Test 1

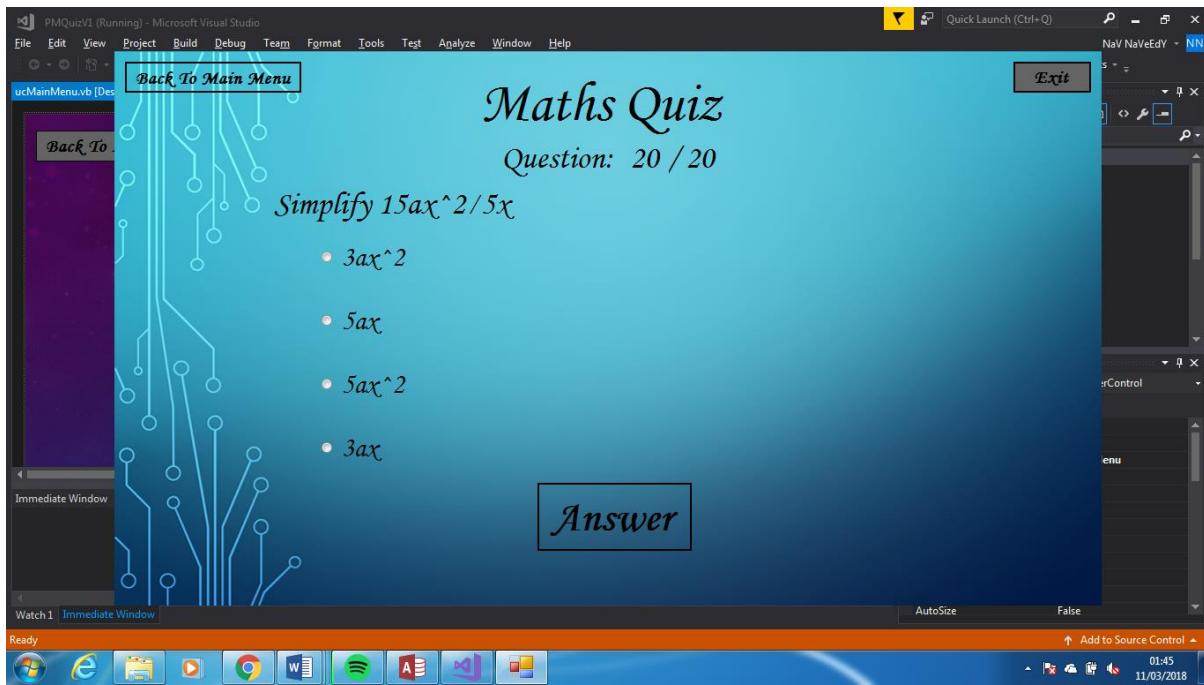


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

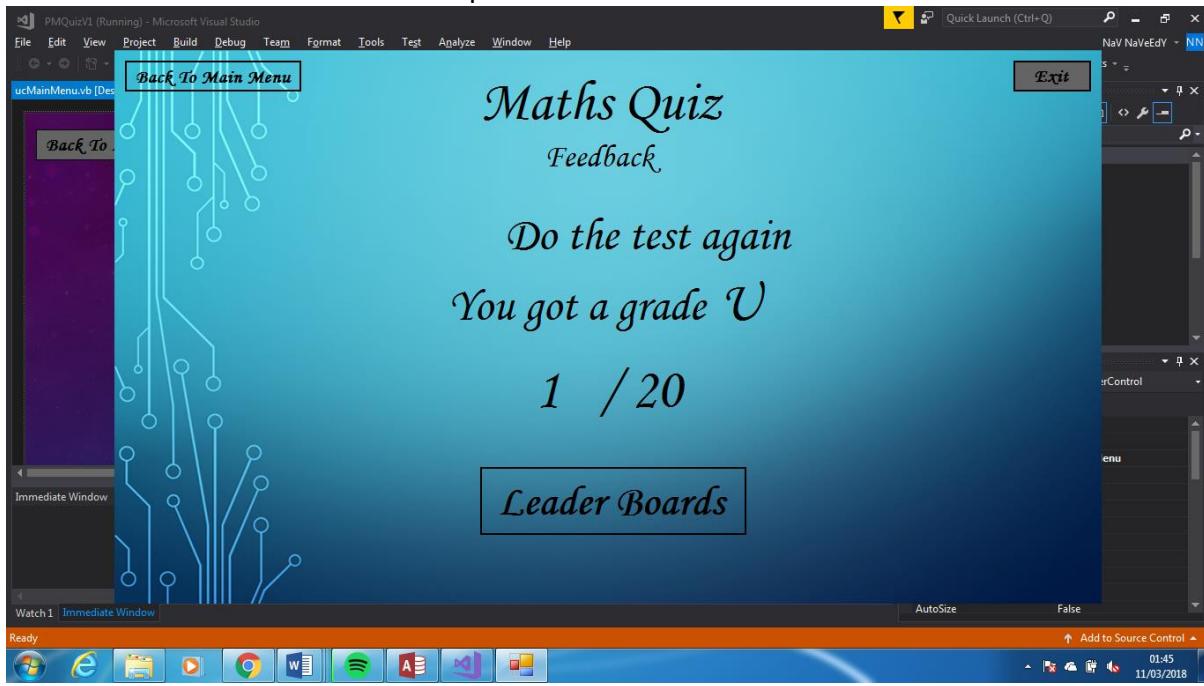


## Test 2

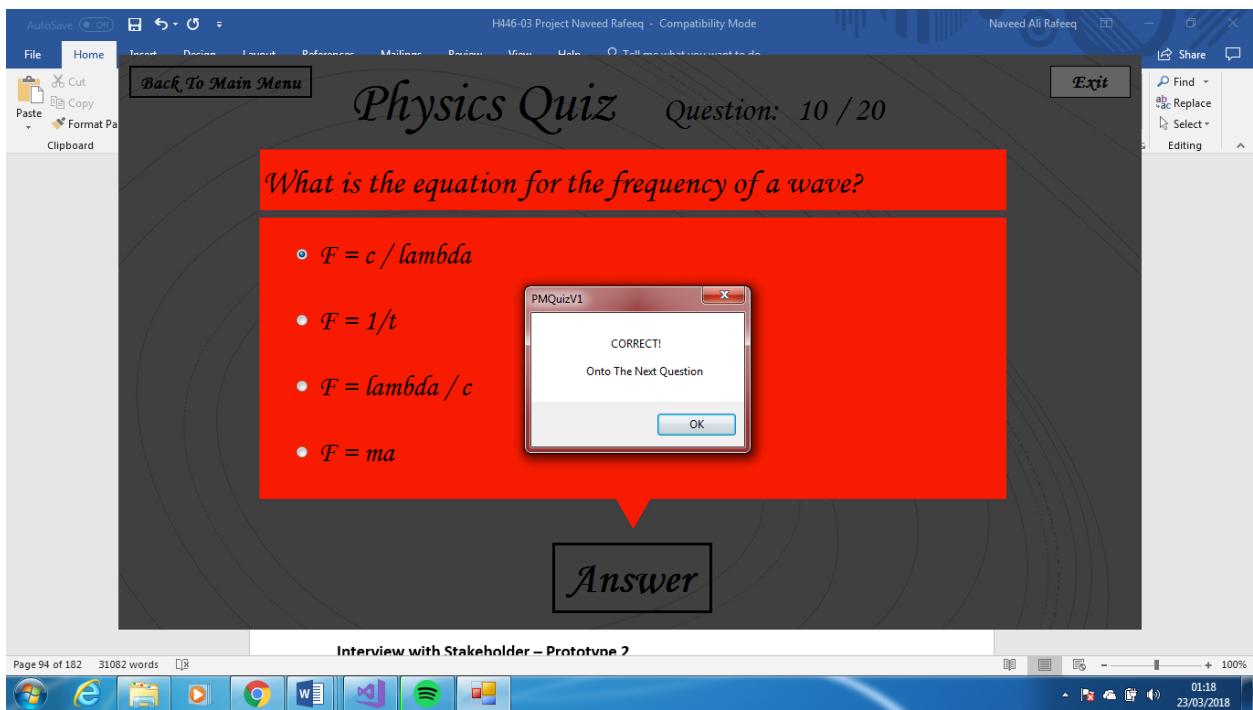


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



### Test 3



Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

A screenshot of Microsoft Word showing a physics quiz. The title "Physics Quiz" is at the top right, and the question "What is the equation for the frequency of a wave?" is in a red box. Below it is a list of four options:

- $F = \lambda / c$
- $F = 1/t$
- $F = ma$
- $F = c / \lambda$

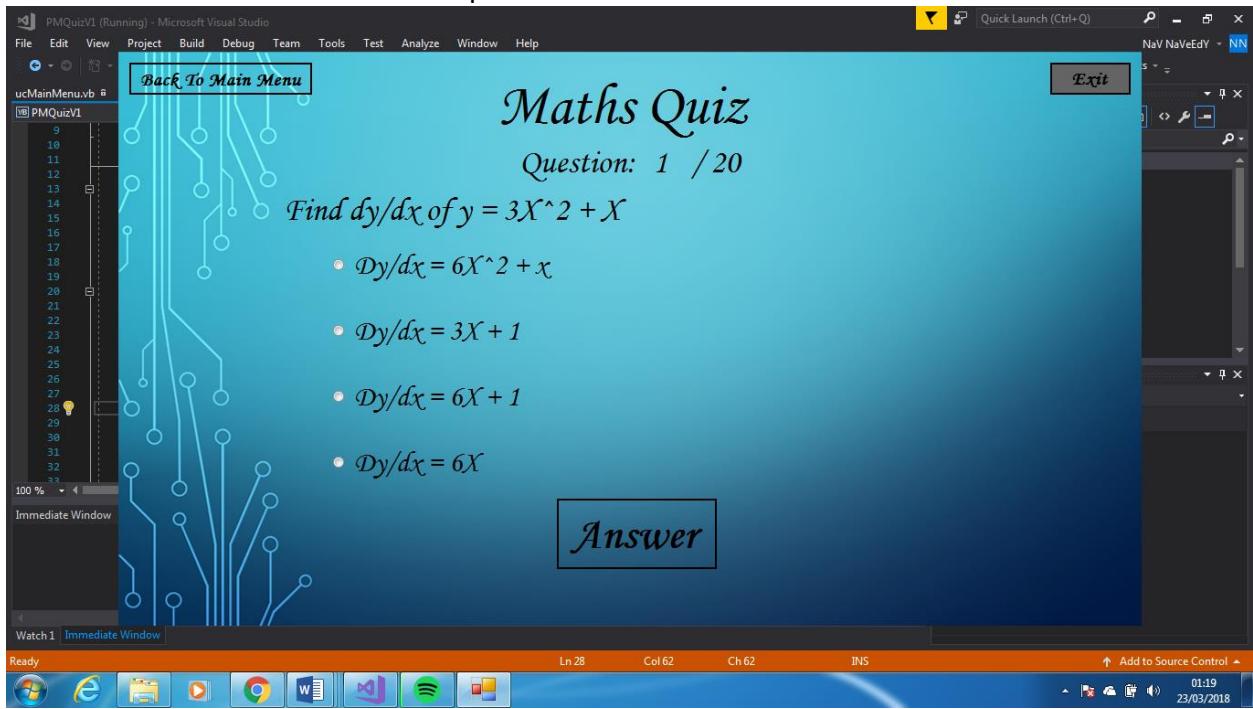
An "Answer" button is visible below the red box. The background features concentric circles.

#### Test 4

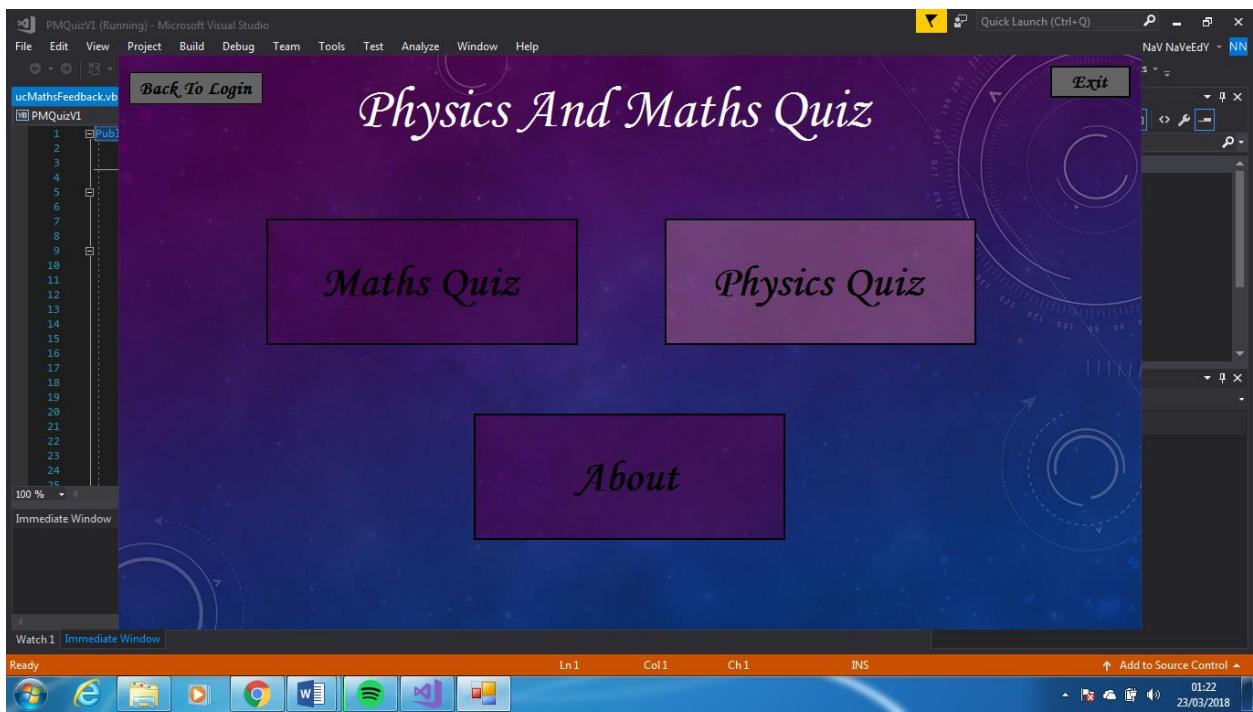
A screenshot of Microsoft Visual Studio showing a "Physics And Maths Quiz" application. The interface includes a sidebar with a navigation menu and three main buttons: "Maths Quiz", "Physics Quiz", and "About". The background has a circular, futuristic design.

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

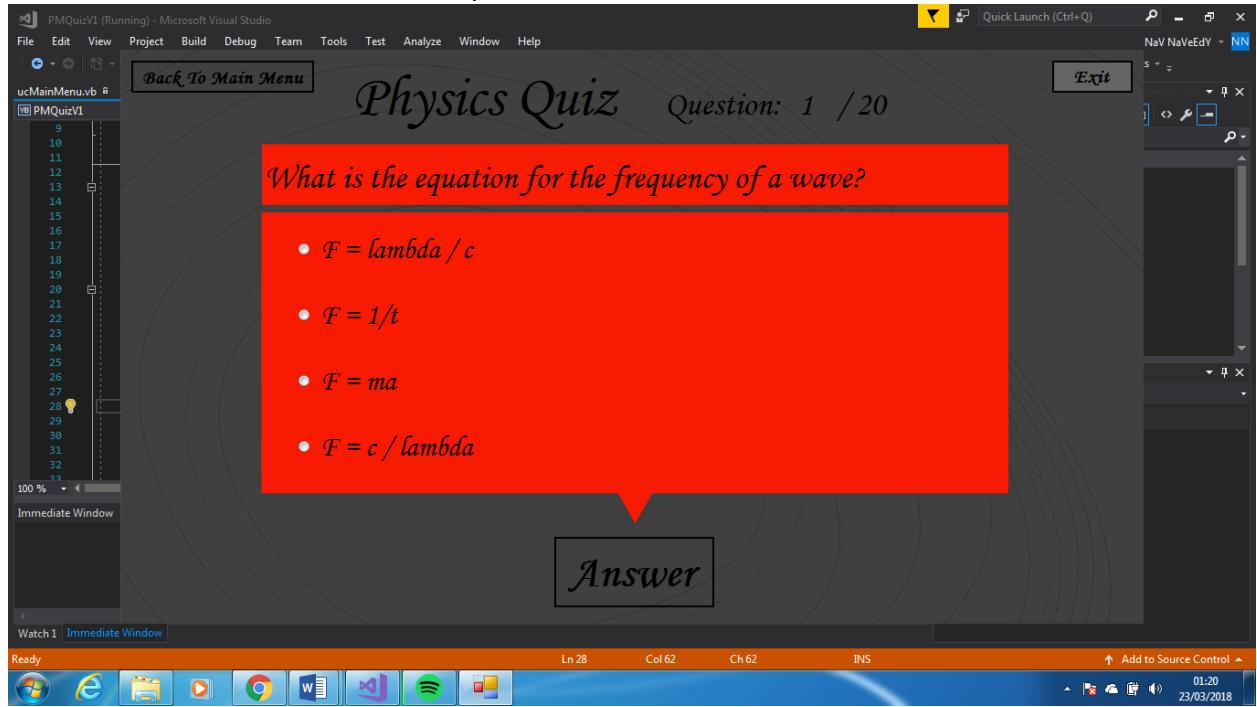


## Test 5

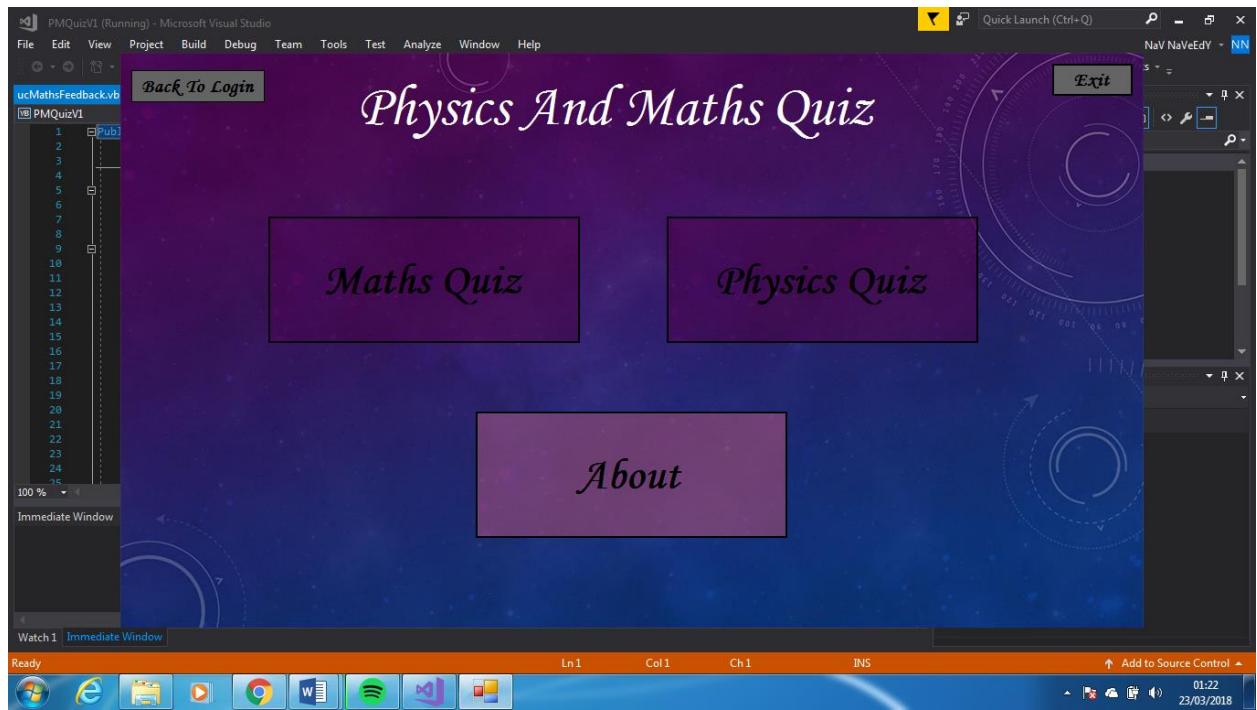


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

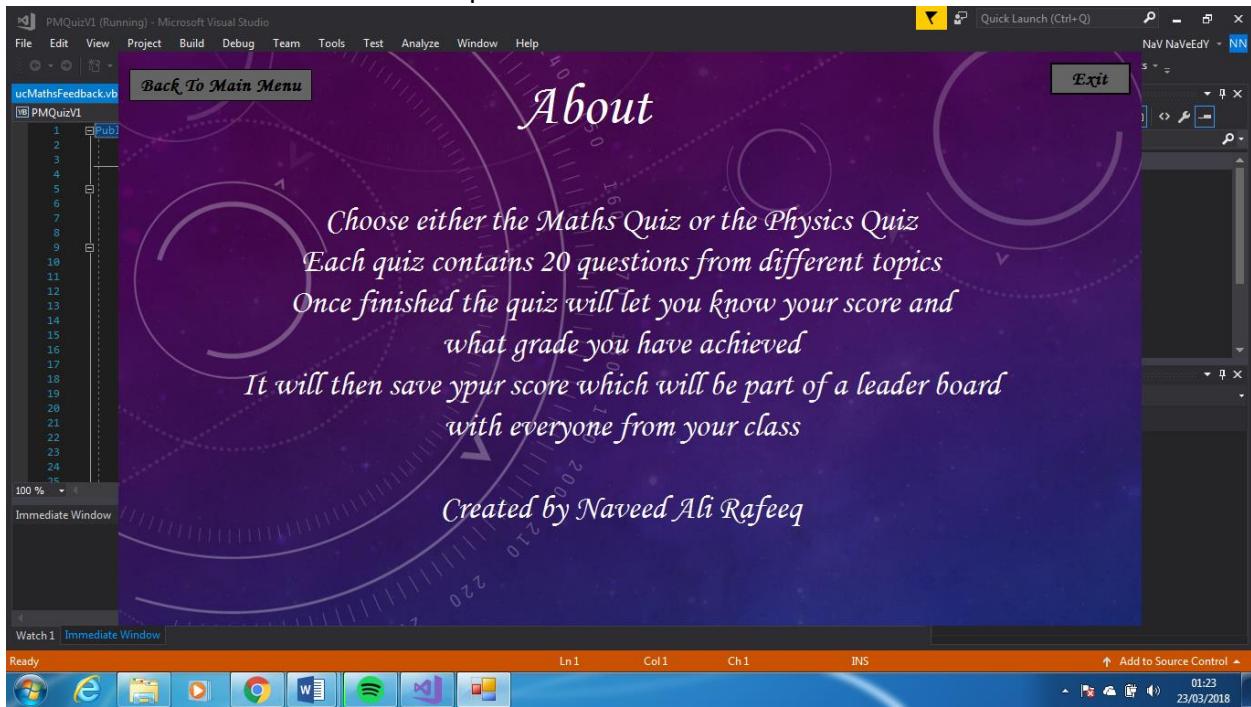


## Test 6

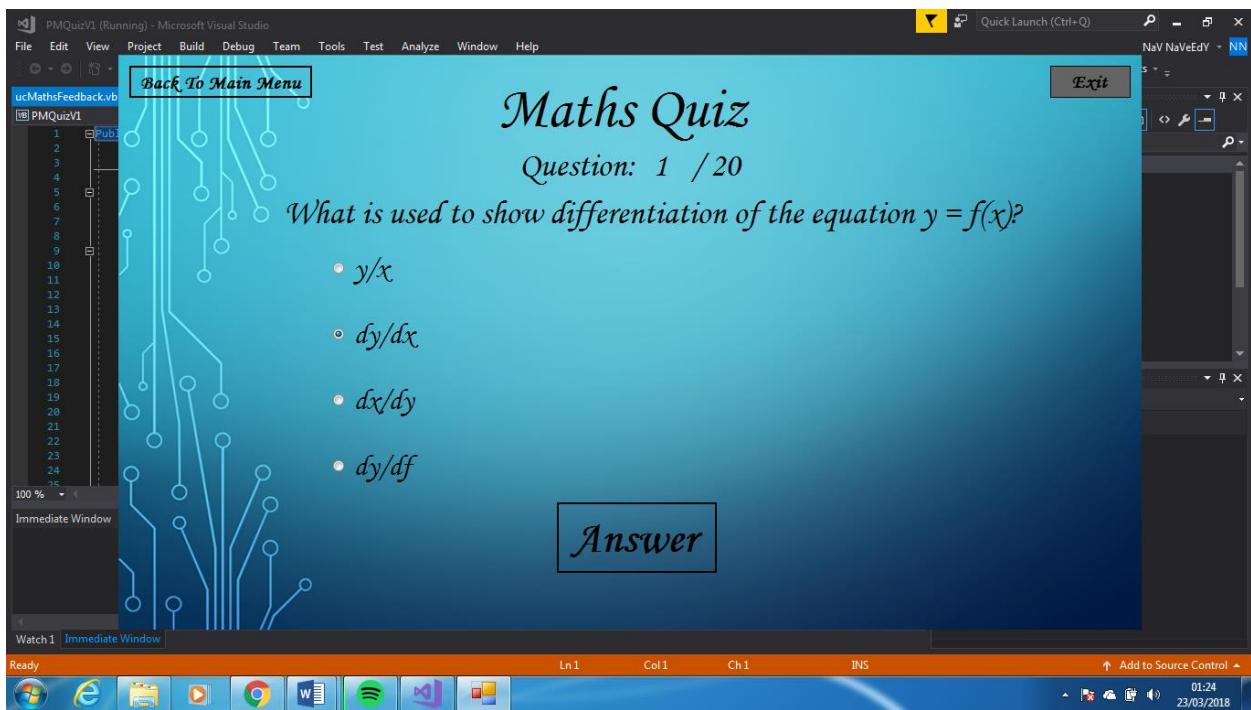


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



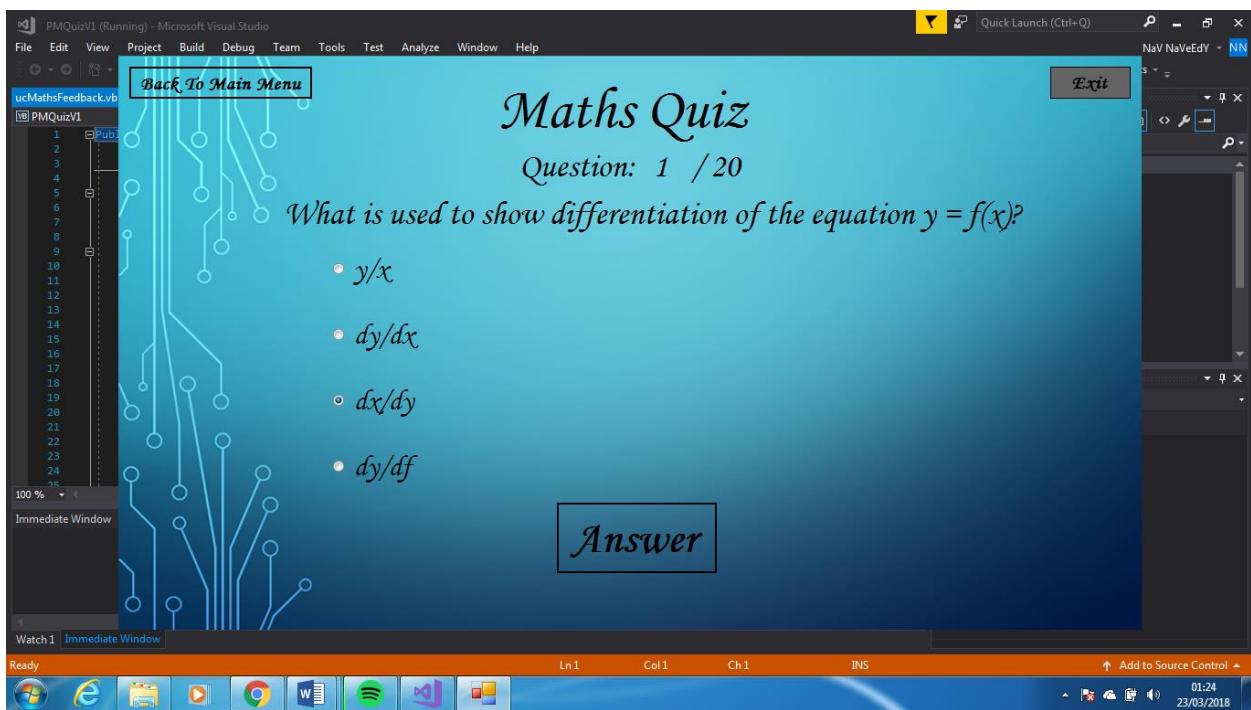
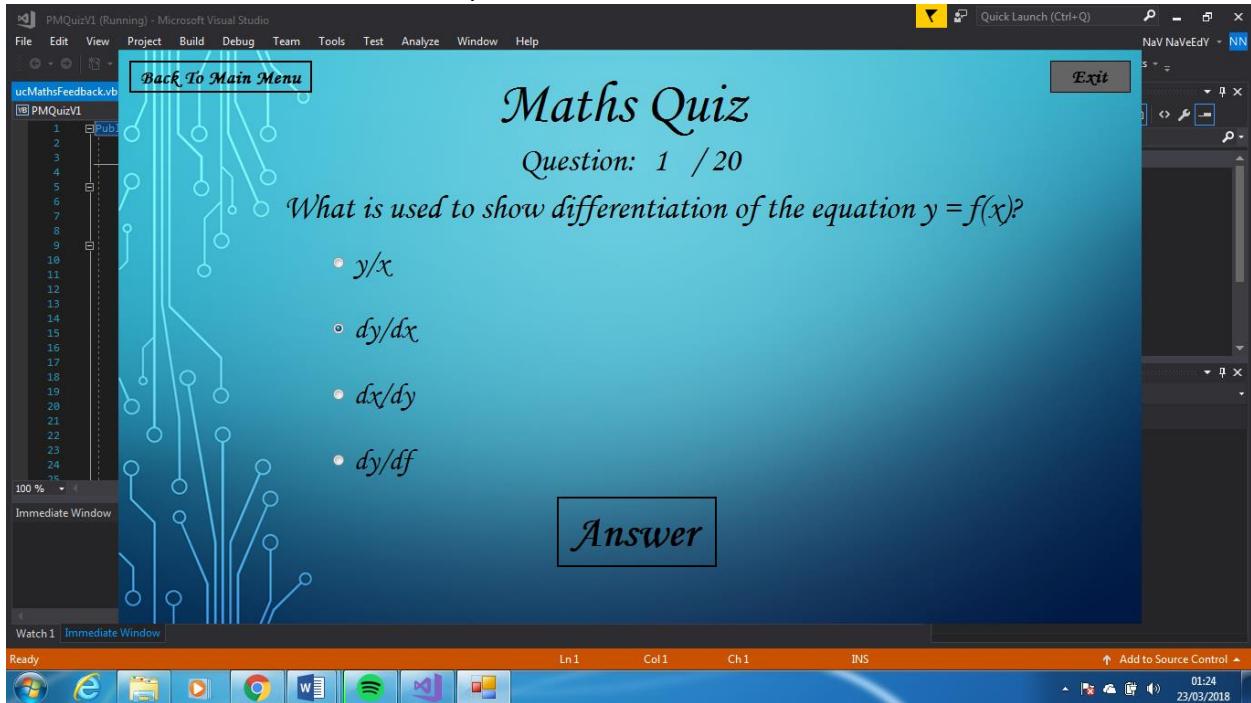
## Test 7



## Test 8

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



## Test 9

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

Back To Main Menu      Physics Quiz      Question: 10 / 20

What is the equation for the frequency of a wave?

- $F = c / \lambda$
- $F = 1/t$
- $F = \lambda / c$
- $F = ma$

PMQuizV1

CORRECT!

Onto The Next Question

OK

Answer

Interview with Stakeholder – Prototype 2

Back To Main Menu      Physics Quiz      Question: 11 / 20

What is the equation for the frequency of a wave?

- $F = \lambda / c$
- $F = 1/t$
- $F = ma$
- $F = c / \lambda$

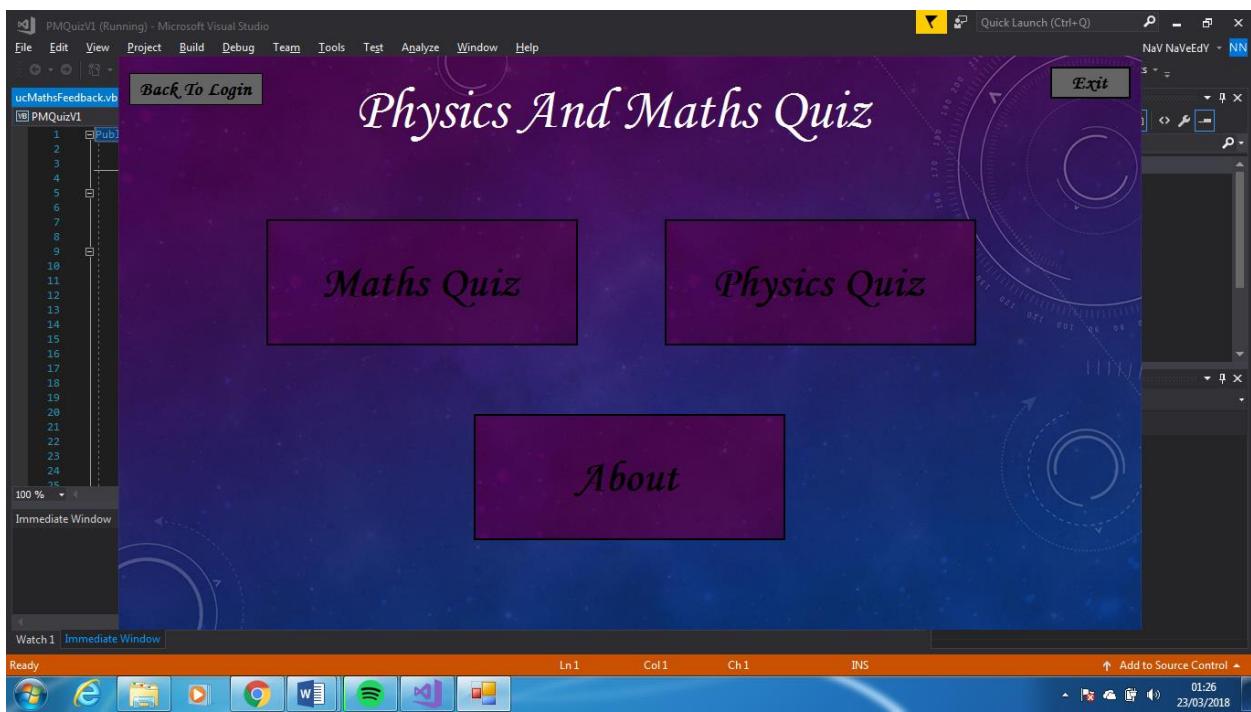
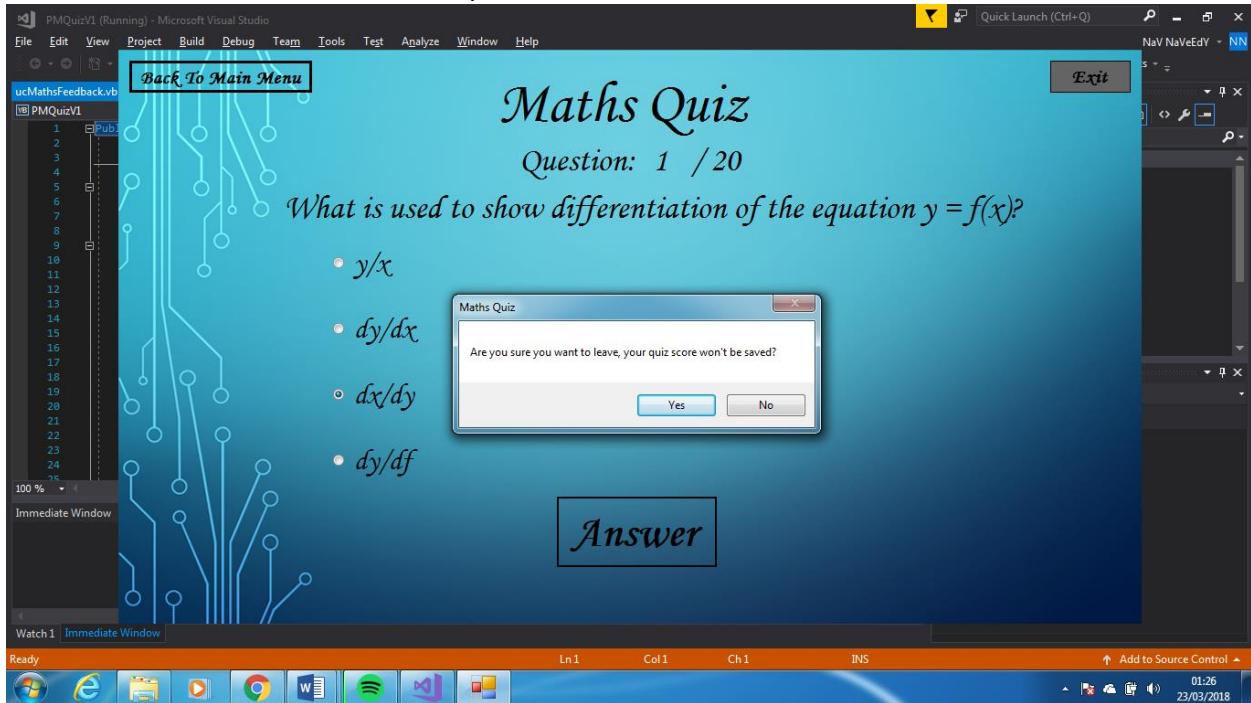
Answer

Interview with Stakeholder – Prototype 2

Test 10

Candidate Name: Naveed Ali Rafeeq

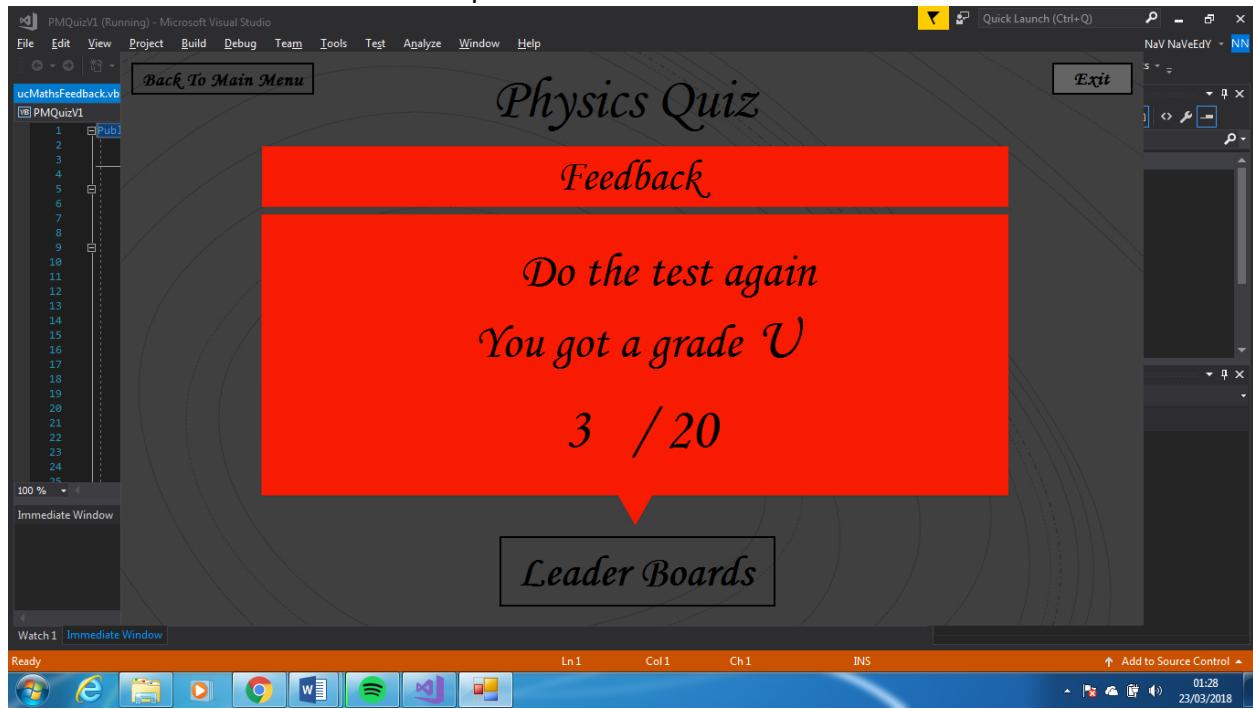
Candidate Number: 1904



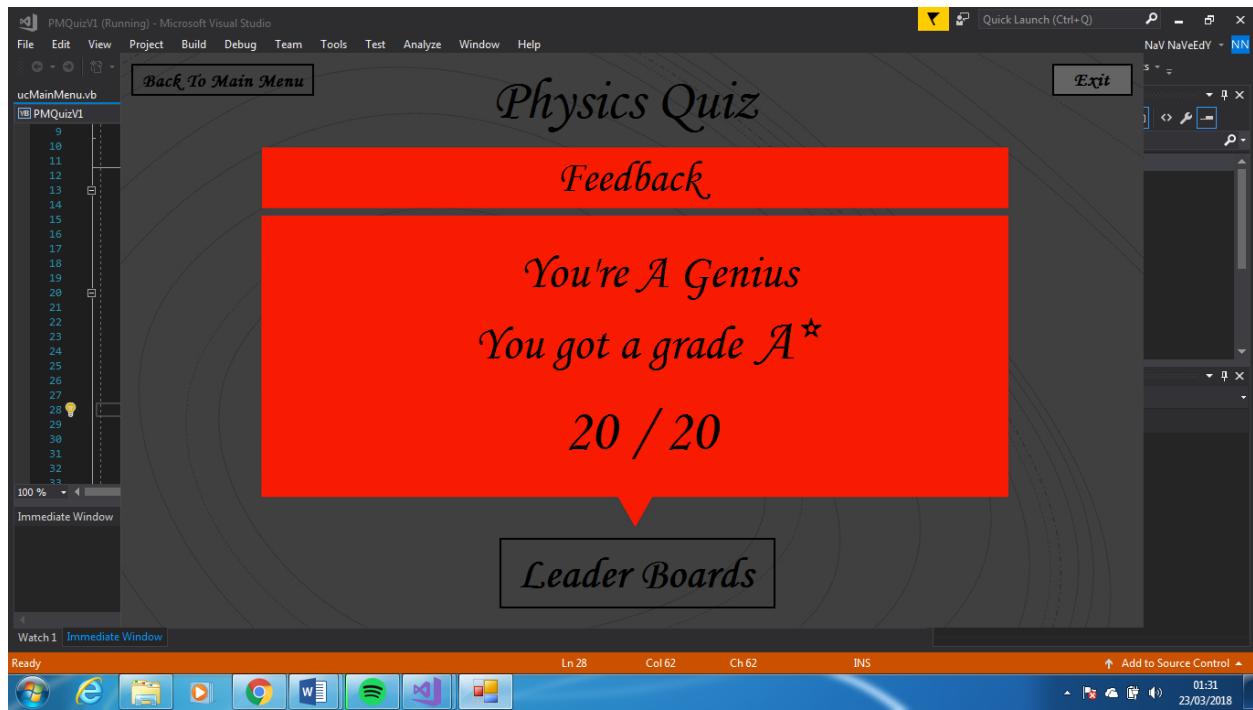
## Test 11

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



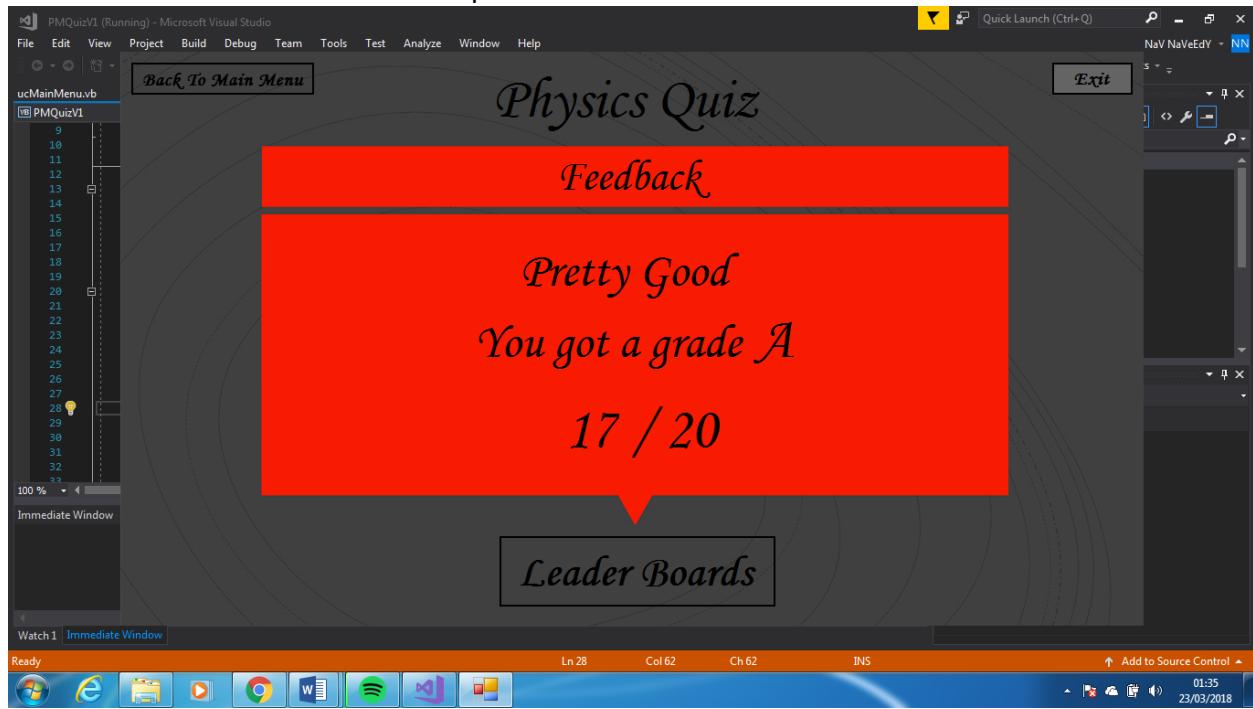
### Test 12



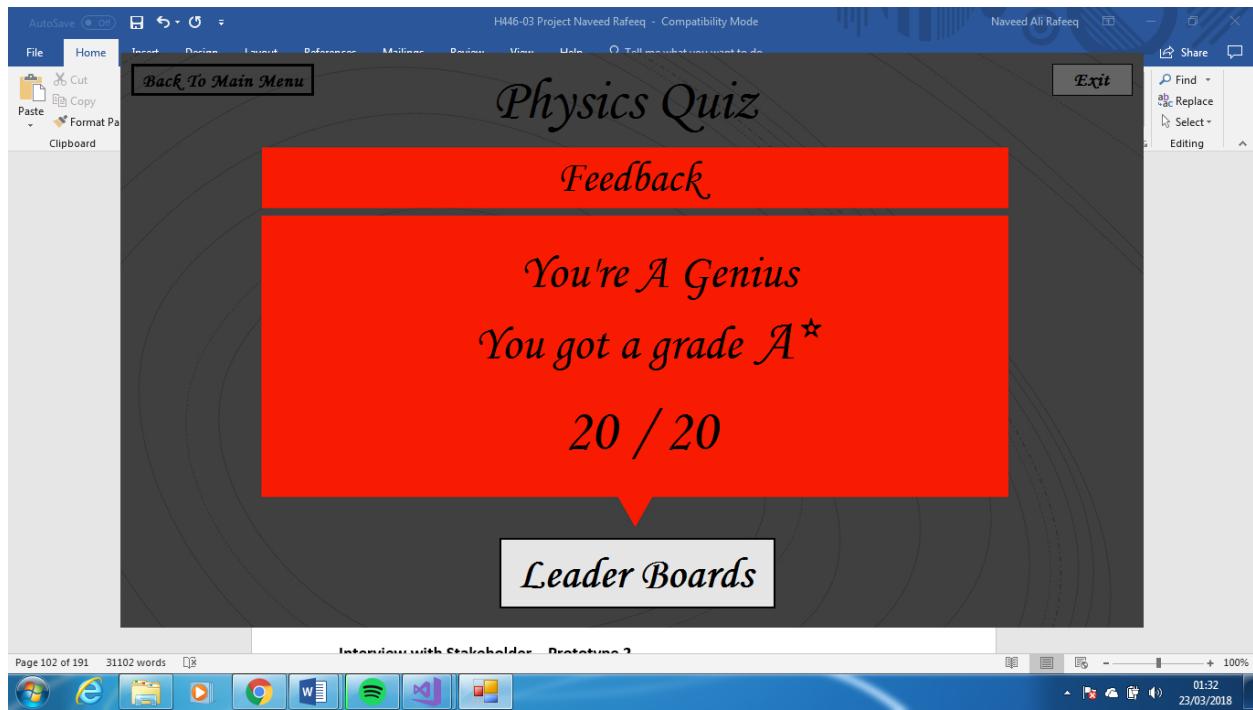
### Test 13

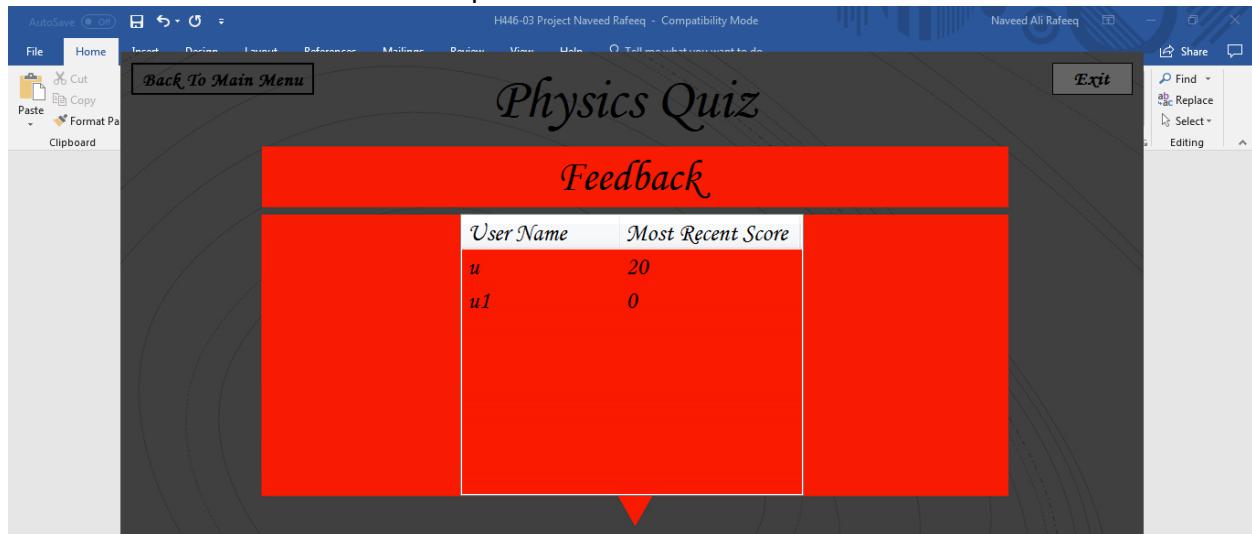
Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



#### Test 14





## Interview with Stakeholder – Prototype 2

I conducted this interview with my other stakeholder Moaz Masood, I decided to do this since this prototype mainly consists of the Quiz and Feedback sections therefore students are going to be using this part of the program much more often. I asked him for his opinions on the program so far and what areas I could improve.

**Me: So, what do you think of the Quiz parts of the program?**

**Moaz:** I think it is quite simple, yet I believe that's a good thing. With Quizzes that's how you want them to be the questions big and bold with your answers clearly labelled underneath. It tells you what question you are currently on as well as a message telling you whether the question is right or wrong.

**Me: Ok that's great, what about the feedback section?**

**Moaz:** The feedback is also quite engaging since its bespoke for each user, it gives you a line of advice and a grade. Also, the leader board creates quite a competitive mood due to everyone's grades being displayed and therefore everyone wants to be as high as possible, no one wants to be at the bottom.

**Me:** Alright thanks that's the intention of the leader board so I'm glad you're impressed by that. Any issues you encountered?

**Moaz:** Yeah, after looking at the leader board I couldn't go back to my feedback, this was frustrating because I forgot what it said and to go back on the feedback I had to do the whole quiz again. This was even more annoying due to the fact that after taking the quiz again I

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

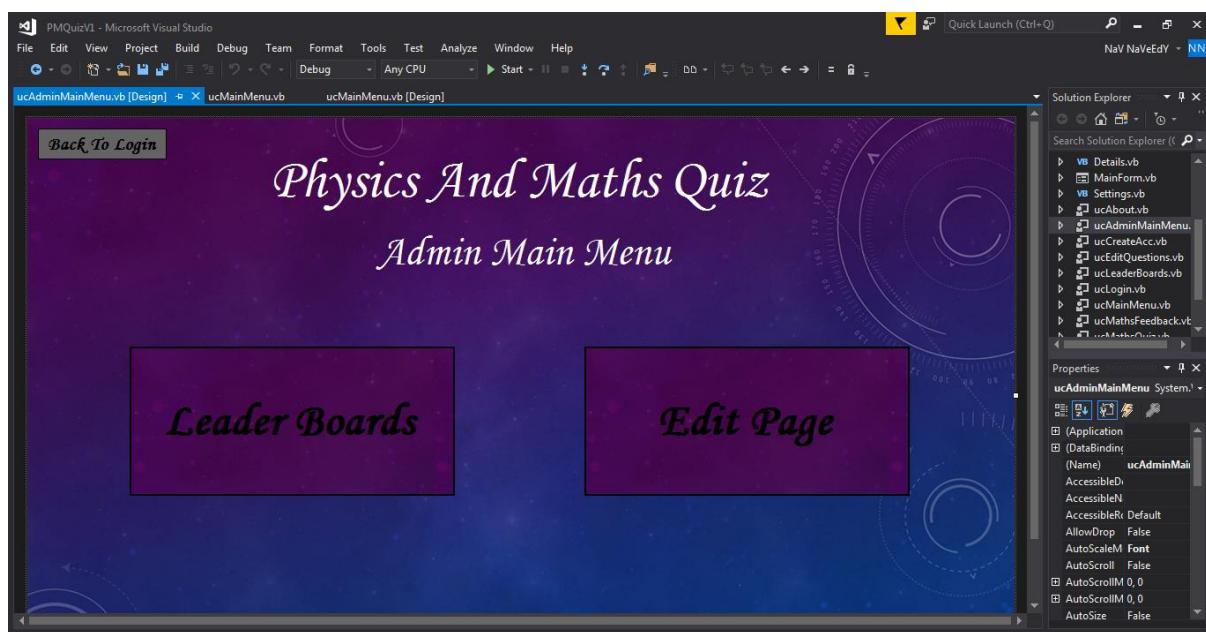
obtained a different grade and got a different line of advice. Also, your leader board size and its font are quite small I would recommend you make it larger.

Me: Ok Moaz thanks for your feedback it really helped.

Moaz: Happy to help.

## Prototype 3

### User Control – Admin Main Menu



The admin main menu consists of 3 buttons, the back to login button, the leader boards button and the edit page. I then used the same opaque images that I used in the main menu to appear when my mouse hovers over the buttons to create the highlight effect again.

```
Private Sub LeadBtn_MouseEnter(sender As Object, e As EventArgs) Handles Leadbtn.MouseEnter
    Leadbtn.BackgroundImage = My.Resources.BgBlurSmall
    'This sub loads a different image when the mouse hovers over the Maths Quiz button
End Sub

Private Sub EditBtn_MouseEnter(sender As Object, e As EventArgs) Handles EditBtn.MouseEnter
    EditBtn.BackgroundImage = My.Resources.BgBlurSmall
    'Same as the above but with the Edit Questions button Quiz button
End Sub
```

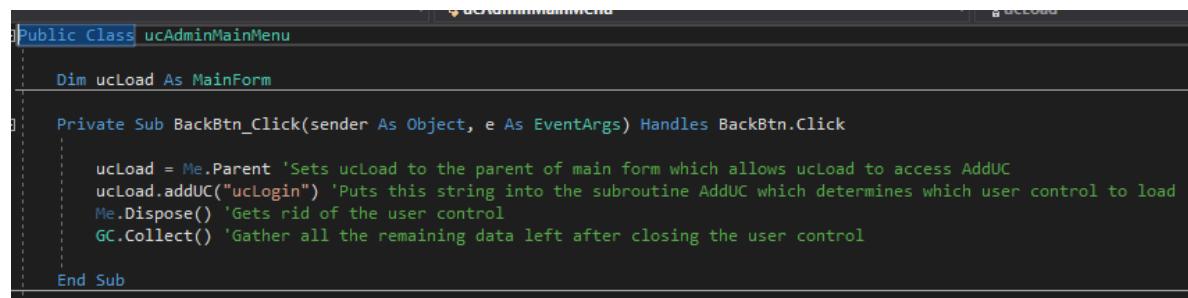
```

Private Sub Leadbtn_Click(sender As Object, e As EventArgs) Handles Leadbtn.Click
    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.addUC("ucLeaderBoards") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control
End Sub

Private Sub EditBtn_Click(sender As Object, e As EventArgs) Handles EditBtn.Click
    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
    ucLoad.addUC("ucEditQuestions") 'Puts this string into the subroutine AddUC which determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control
End Sub
End Class

```

The 2 subs above run when you click their corresponding button, this means that they direct you to other User Controls within the program.



```

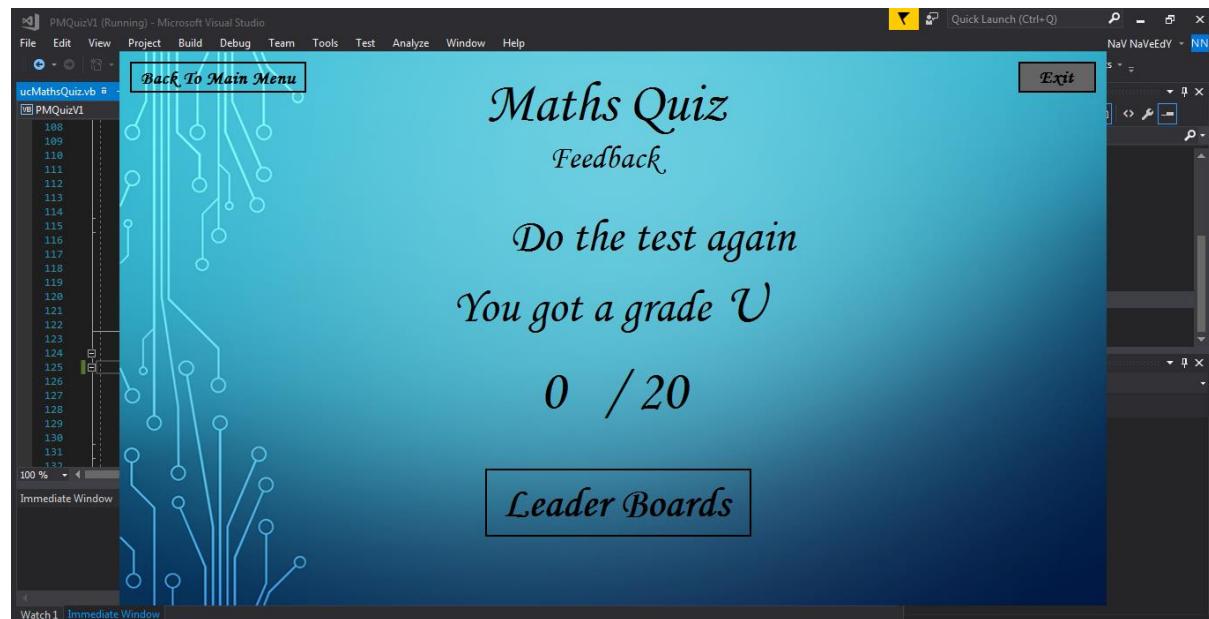
Public Class ucAdminMainMenu
    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    End Sub

```

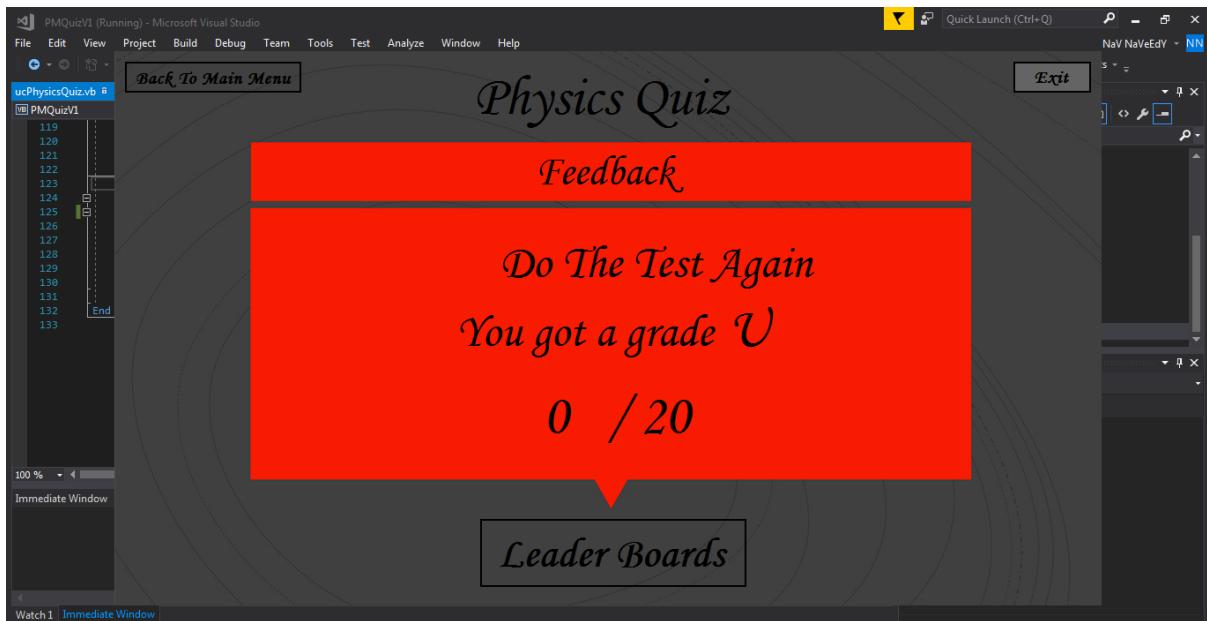
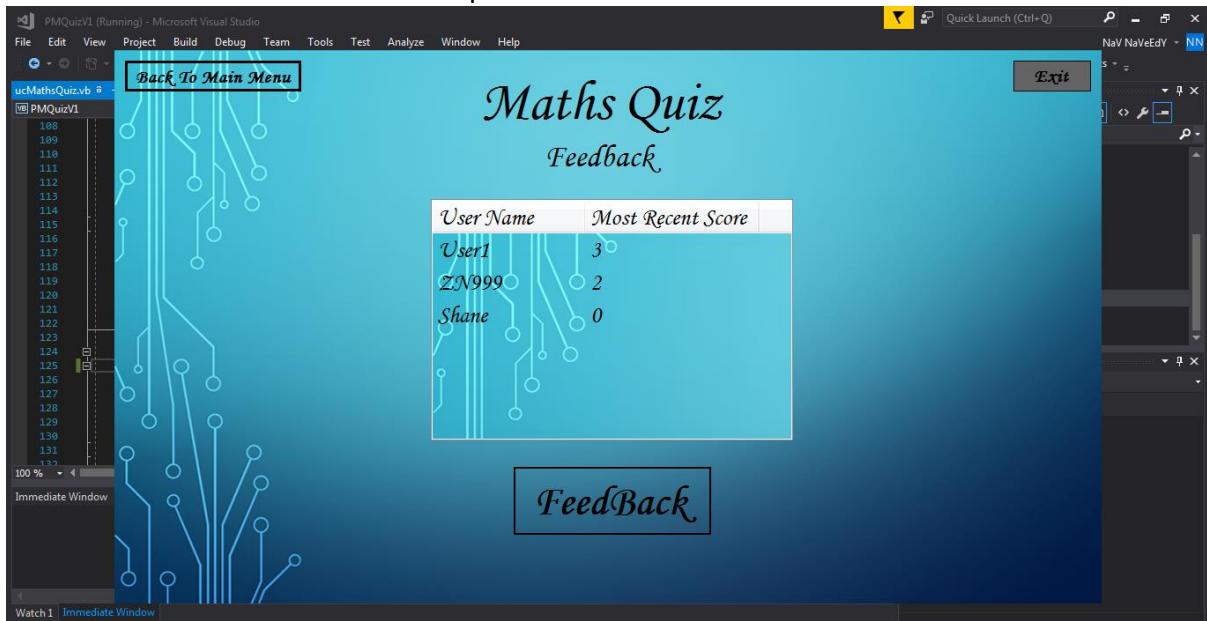
And obviously you have the back button that takes you back to the Login UC by disposing of the current UC and adding the Login UC.

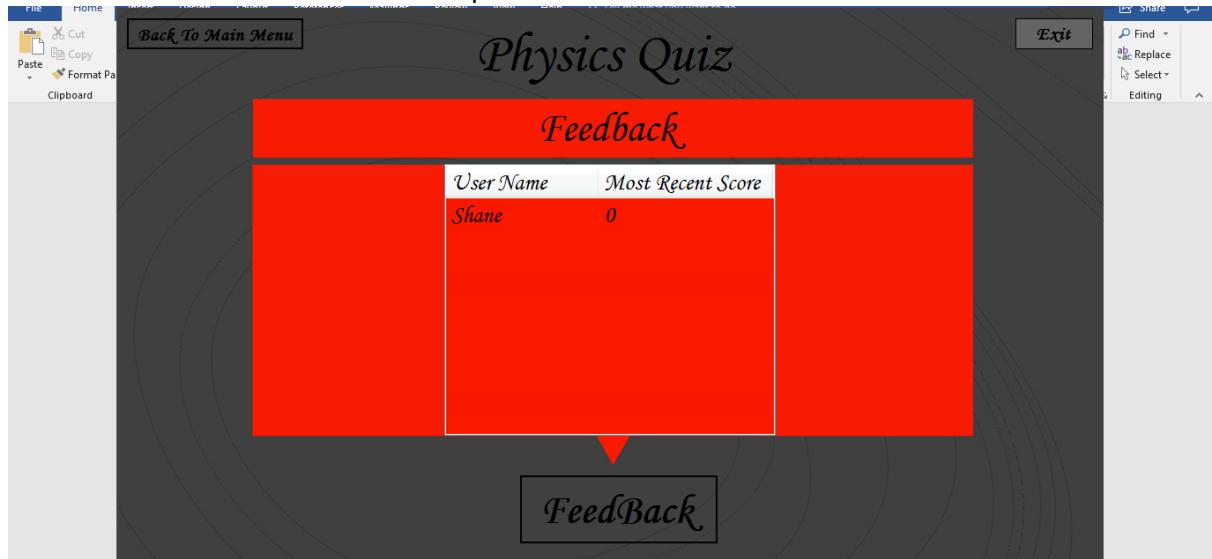
## Amendments to the Feedback User Controls



Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904





I increased the sizes of the list views as well as the fonts of them so that the leader boards now stand out more.

You can also now go back to the feedback that was given to you.

```

Private Sub FeedBtn_Click(sender As Object, e As EventArgs) Handles FeedBtn.Click
    'When the Leader Boards button is clicked the sub hides the listview and the Feedback button,
    'then makes the Feedback text and the Leader Board button visible
    LstScore.Visible = False
    FBackLbl.Visible = True
    Label2.Visible = True
    GrdLbl.Visible = True
    QtnRLbl.Visible = True
    Label3.Visible = True
    FeedBtn.Visible = False
    LeadBtn.Visible = True

    End Sub
End Class

```

I created another button called Feedback. This button does the opposite of the Leader Boards button, so it hides the list view and the Feedback button, and it displays all of the feedback labels.

```

LstScore.Visible = False 'The list view is hidden
FeedBtn.Visible = False 'The button to send you back to the feedback is hidden

```

Due to the new Feedback button the second line is now added to the sub that runs when the UC loads.

```

Private Sub LeadBtn_Click(sender As Object, e As EventArgs) Handles LeadBtn.Click
    'When the Leader Boards button is clicked the sub hides all the feedback text and the Leader board
    'button, then makes the listview and the Feedback button visible
    LstScore.Visible = True
    FBackLbl.Visible = False
    Label2.Visible = False
    GrdLbl.Visible = False
    QtnRLbl.Visible = False
    Label3.Visible = False
    FeedBtn.Visible = True
    LeadBtn.Visible = False

End Sub

```

Because of the Feedback button the 2 lines at the bottom are now added in order to display the Feedback button when the Leader Boards button is clicked, and the Leader Boards button is now hidden.

## Amendment To Login User Control

So, to get from the Login to the Admin Main Menu I added to the IF statement that I originally used to load the main menu. Before, if the text within the login textboxes matched those within the either the Student or Admin Tables then the Main Menu would load. However now it does the following.

```

Private Sub LoginBtn_Click(sender As Object, e As EventArgs) Handles LoginBtn.Click
    Dim dtAdmin, dtStudent As New DataTable 'Creates 2 datatable variables
    dtAdmin = runSQL("Select * from AdminData where TUserN = '" & UsrNmeTxt.Text & "' and TPass = '" & PassTxt.Text & "'")
    dtStudent = runSQL("Select * from StudentData where SUserN = '" & UsrNmeTxt.Text & "' and SPass = '" & PassTxt.Text & "'")
    'Extracts all the Usernames and Passwords from the Admin and Student Tables

    If dtStudent.Rows.Count = 1 Then
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucMainMenuItem") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
        Details.LoginUser = dtStudent.Rows(0)(0) 'Saves the SUserN on the student row to be used later
        Details.UserID = dtStudent.Rows(0)(2) 'Saves the TID on the student row to be used later
    ElseIf dtAdmin.Rows.Count = 1 Then
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucAdminMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
        Details.LoginUser = dtAdmin.Rows(0)(1) 'Saves the TUserN on the admin row to be used later
        Details.UserID = dtAdmin.Rows(0)(0) 'Saves the TID on the admin row to be used later
    Else
        MsgBox("Your details have been entered incorrectly, please try again")
    End If

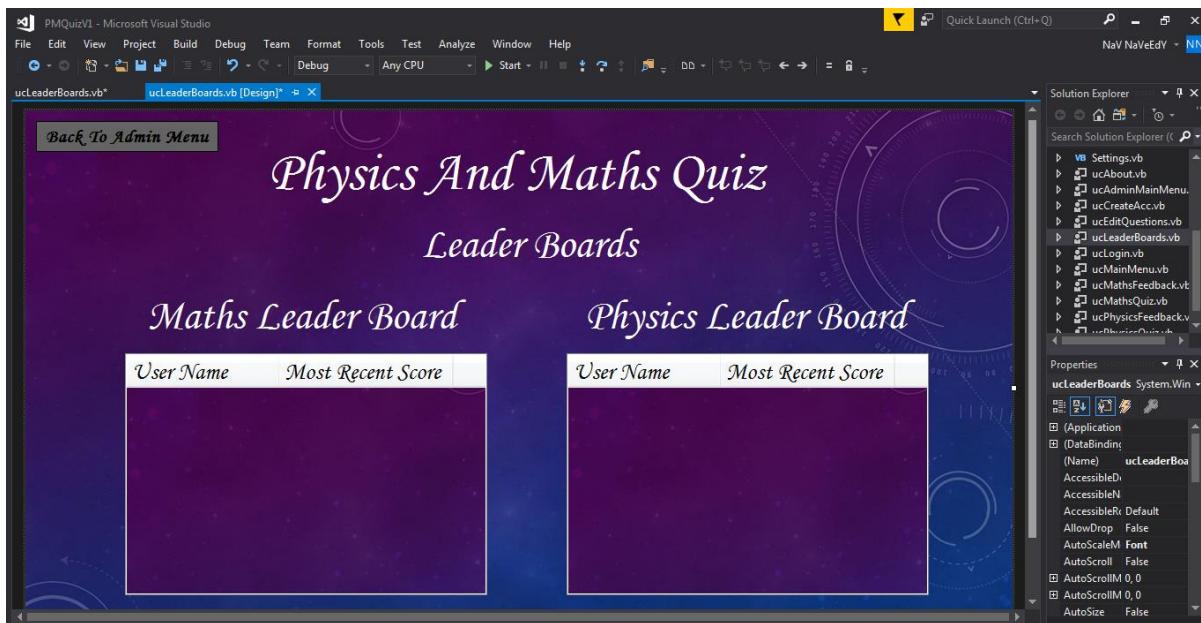
End Sub

```

What this IF statement does now is load the Main Menu if the details within the textboxes matches those within the Student Table. It then saves their name and Teacher ID to global variables so that they can be used in other classes at a later time. If the textboxes match with those in the Admin Table, then it will load a different User Control the Admin Main Menu. Therefore, it will only be accessible to the teachers this is due to the fact that it has the ability to edit questions which should only be accessible by the teachers and not the students. It also

Candidate Name: Naveed Ali Rafeeq Candidate Number: 1904  
 saves the teachers Login and their Teacher ID into global variables as well. This can then be used later in order to load leader boards from only their class and thus they can only delete the users within their class as well.

## User Control – Leader Boards



The user control consists of 1 button and 2 list views. As shown one list view displays the username and score of the Maths Results table that have the same corresponding Teacher ID and the other for the Physics Results table.

```
Public Class ucLeaderBoards
    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucAdminMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    End Sub

```

Sub that runs when the Back button is clicked that loads the Admin Main Menu UC.

```

Private Sub MathsScoreLoad() 'Sub that places all of the Usernames and their adjacent Maths scores into the List View

    Dim dt As New DataTable 'Copies a table within the database into vb to be edited and then updated
    dt = runSQL("SELECT * from MResults where TID = '" & Details.UserID & "' ORDER BY Results DESC;")
    'Loads all the rows that have the same TID as the one used to log in and places them into the Data Table

    For x = 0 To dt.Rows.Count - 1
        Dim items(1) As String '1 dimensional array with 2 values
        items(0) = dt.Rows(x)(0) 'Username
        items(1) = dt.Rows(x)(1) 'Score
        'Adds the username and password into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items array into the LineNew
        MLstScore.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub

```

I then created this sub that loads all of the items in the Maths Results table into the Maths Leader board, the left list view. This sub is identical to the one used in the Maths Feedback User Control.

```

Private Sub PhysicsScoreLoad() 'Sub that places all of the Usernames and their adjacent Physics scores into the List View

    Dim dt As New DataTable
    dt = runSQL("SELECT * from PResults where TID = '" & Details.UserID & "' ORDER BY Results DESC;")
    'Loads all the rows that have the same TID as the one used to log in and places them into the Data Table

    For x = 0 To dt.Rows.Count - 1
        Dim items(1) As String '1 dimensional array with 2 values
        items(0) = dt.Rows(x)(0) 'Username
        items(1) = dt.Rows(x)(1) 'Score
        'Adds the username and password into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items array into the LineNew
        PLstScore.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub

```

This sub is the same but for the right list view or the Physics Leader Board and it fills it with the items within the Physics Results table.

```

Private Sub ucLeaderBoards_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'When the UC loads it runs the two subs that places the items within the leaderboards

    MathsScoreLoad()
    PhysicsScoreLoad()

End Sub
End Class

```

So, when the UC loads so does this sub and all it does is call upon the other 2 subs that fills the Leader Boards.

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

The screenshot shows a Windows application window titled "Physics And Maths Quiz". At the top left is a menu bar with options: File, Project, Build, Debug, Team, Format, Tools, Test, Analyze, Window, Help. On the right side are "Back To Admin Menu" and "Exit" buttons. The main area contains two tables:

User Name	Most Recent Score
u	19
NaveedR	17
u1	0
Moaz Masood	0

User Name	Most Recent Score
u	1
u1	0

Below the tables, there are "AutoSize" and "False" buttons. The background features a circular dial with numbers from 0 to 360.

### User Control – Edit Questions

The screenshot shows a Windows application window titled "Physics And Maths Quiz". At the top left is a menu bar with options: File, Project, Build, Debug, Team, Format, Tools, Test, Analyze, Window, Help. On the right side are "Back To Admin Menu" and "Exit" buttons. The main area contains three buttons:

- A large red button labeled "Delete Users".
- A blue button labeled "Edit Maths Questions".
- A blue button labeled "Edit Physics Questions".

Below the buttons, there are "AutoSize" and "False" buttons. The background features a circular dial with numbers from 0 to 360.

This is what is displayed when the UC is loaded. With this User Control I used many subs just to hide and show different list views, text boxes and buttons.

```

Public Class ucEditQuestions
    Dim ucLoad As MainForm
    Dim DeleteUser As String 'The Username of the account the admin wishes to delete

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
        ucLoad.addUC("ucAdminMainMenu") 'Puts this string into the subroutine AddUC which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    End Sub

```

Back to the Admin Main Menu button.

```

Private Sub UsrLstBtn_MouseEnter(sender As Object, e As EventArgs) Handles UsrLstBtn.MouseEnter
    UsrLstBtn.BackgroundImage = My.Resources.BgBlurSmall
    'This sub loads a different image when the mouse hovers over the Delete User button
End Sub

Private Sub MQtnBtn_MouseEnter(sender As Object, e As EventArgs) Handles MQtnbtn.MouseEnter
    MQtnbtn.BackgroundImage = My.Resources.BgBlurSmall
    'Same as the above but with the Edit Maths Questions button
End Sub

Private Sub PQtnBtn_MouseEnter(sender As Object, e As EventArgs) Handles PQtnbtn.MouseEnter
    PQtnbtn.BackgroundImage = My.Resources.BgBlurSmall
    'Same as the above but with the Edit Physics Questions button
End Sub

```

Same opaque backgrounds for each main button in the form.

When the Delete Users button is pressed it triggers a number of subs:

```

Private Sub UsrLstBtn_Click(sender As Object, e As EventArgs) Handles UsrLstBtn.Click
    'Sub that loads the User List when the Delete User button is clicked

    UsrLstShow() 'Sub that loads the User List View and its delete button
    BtnsHide() 'Sub that hides the menu buttons
    EditBtn.Visible = True 'Edit button is visible
    BackBtn.Visible = False 'Back button is hidden
End Sub

```

```

Private Sub UsrLstShow() 'Sub that loads the User List and its delete user button
    UserLst.Visible = True
    DeleteUserBtn.Visible = True
    UserList()
End Sub

```

```

Private Sub BtnsHide() 'Sub that hides the menu buttons
    UsrLstBtn.Visible = False
    MQtnbtn.Visible = False
    PQtnbtn.Visible = False
End Sub

```

```

Private Sub UserList() 'Sub that loads all students in with matching TID's as the admin

    Dim dt As New DataTable 'Copies a table within the database into vb to be edited and then updated

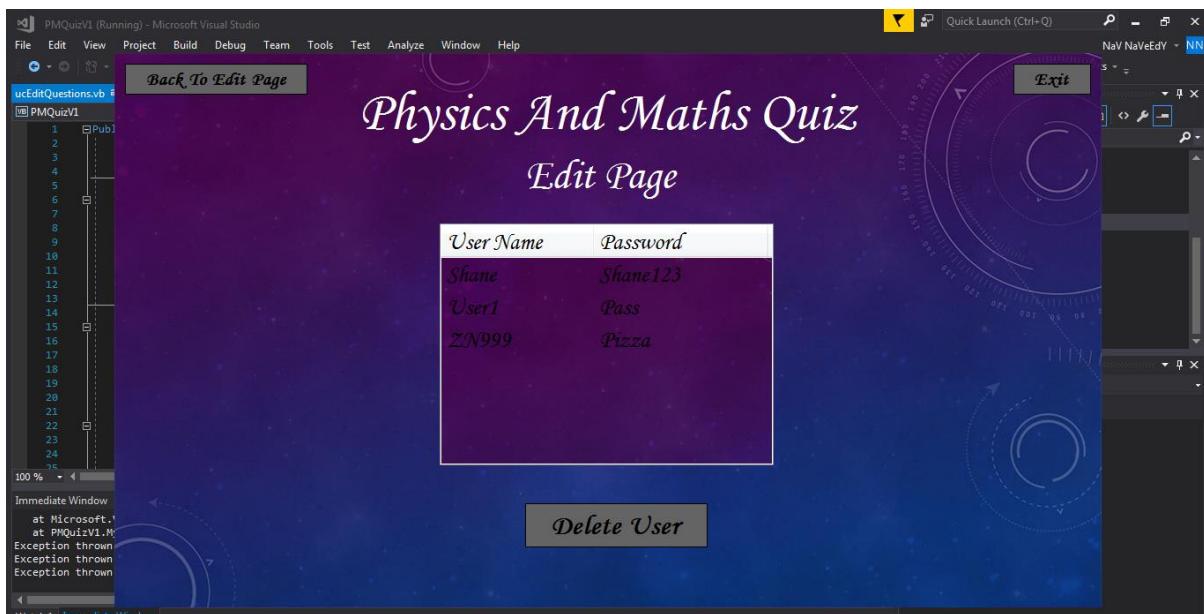
    dt = runSQL("SELECT SUserName, SPass from StudentData where TID = '" & Details.UserID & "' ORDER BY SUserName ASC;")
    'Loads all the users with their usernames and passwords into the DataTable

    For x = 0 To dt.Rows.Count - 1
        Dim items(1) As String '1 dimensional array with 2 values
        items(0) = dt.Rows(x)(0) 'Username
        items(1) = dt.Rows(x)(1) 'Password
        'Adds the username and password into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items array into the LineNew
        UserList.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub

```

To simplify, the button will display the list view containing all Admins students and the delete button that goes with it. It also hides all the buttons on the last page and changes the back to Admin main menu button into the back to edit page button.



Here the admin selects a user and clicks the delete the button.

```

Private Sub UsrLst_SelectedIndexChanged(sender As Object, e As EventArgs) Handles UserLst.Click
    DeleteUser = UserLst.SelectedItems(0).SubItems(0).Text 'The selected users username is now equal to the variable
End Sub

```

```

Public Class ucEditQuestions

    Dim ucLoad As MainForm
    Dim DeleteUser As String 'The Username of the account the admin wishes to delete

```

So whatever username the user selects it makes the Delete User variable equal to it, this is how it deletes the user selected.

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles DeleteUserBtn.Click
    'Button that deletes everything of the user selected

    runSQL("DELETE * FROM StudentData WHERE SUserName = '" & DeleteUser & "'") 'Deletes whole row from StudentData table
    runSQL("DELETE * FROM MResults WHERE SUserName = '" & DeleteUser & "'") 'Deletes whole row from MResults table
    runSQL("DELETE * FROM PResults WHERE SUserName = '" & DeleteUser & "'") 'Deletes whole row from PResult table
    MsgBox("User has been deleted")

    'Updates the List View
    UserLst.Items.Clear() 'Clear all the items from the List View
    UserList() 'Loads all the items again
End Sub

```

What the sub does is not only delete everything from the Student table but deletes all the users details from the Results tables as well so that their scores are no longer there since they are no longer part of the teacher's class. Teachers need this sub due to many reasons like if they user has left the college or if they made a username that is not deemed suitable by the teacher.

```

Private Sub EditBtn_Click(sender As Object, e As EventArgs) Handles EditBtn.Click
    'Sub that hides all the list views and makes the menu buttons visible when the Edit button is clicked

    BtnsShow() 'Sub that makes the menu buttons visible
    UsrlstHide() 'Sub that hides the User List View and its delete button
    MQtnHide() 'Sub that hides the Maths Question List View, the text boxes and its buttons
    PQtnHide() 'Sub that hides the Physics Question List View, the text boxes and its buttons
    BackBtn.Visible = True 'Back button is visible
    EditBtn.Visible = False 'Edit button is hidden
    TitleLbl.Location = New Point(263, 27) 'Moves the Title back to its original position
End Sub

```

When you click the back to edit page button it runs this sub as well as these:

```

Private Sub BtnsShow() 'Sub that loads the menu buttons
    UsrlstBtn.Visible = True
    MQtnbtn.Visible = True
    PQtnbtn.Visible = True
End Sub

```

```

Private Sub UsrlstHide() 'Sub that hides the User List and its delete user button
    UserLst.Visible = False
    DeleteUserBtn.Visible = False
    UserLst.Items.Clear()
End Sub

```

```

Private Sub MQtnHide() 'Sub that hides the Maths Question List View, the text boxes and its buttons
    MQtnLst.Visible = False
    Qtntxt.Visible = False
    A1txt.Visible = False
    A2txt.Visible = False
    A3txt.Visible = False
    A4txt.Visible = False
    QtnLbl.Visible = False
    A1Lbl.Visible = False
    A2Lbl.Visible = False
    A3Lbl.Visible = False
    A4Lbl.Visible = False
    MRefreshBtn.Visible = False
    MQtnAddBtn.Visible = False
    MQtnDeleteBtn.Visible = False
    MQtnLst.Items.Clear() 'Clears all the items from the List View so that duplicate values will not
    'be there when it is next loaded
End Sub

```

```

Private Sub PQtnHide() 'Sub that hides the Physics Question List View, the text boxes and its buttons
    PQtnLst.Visible = False
    Qtntxt.Visible = False
    A1txt.Visible = False
    A2txt.Visible = False
    A3txt.Visible = False
    A4txt.Visible = False
    QtnLbl.Visible = False
    A1Lbl.Visible = False
    A2Lbl.Visible = False
    A3Lbl.Visible = False
    A4Lbl.Visible = False
    PRefreshBtn.Visible = False
    PQtnAddBtn.Visible = False
    PQtnDeleteBtn.Visible = False
    PQtnLst.Items.Clear() 'Clears all the items from the List View so that duplicate values will not
    'be there when it is next loaded
End Sub

```

What this button does is it calls upon subs that hides all of the list views and its counterparts so that no matter which one was visible they all are hidden after clicking the button. Then it calls upon the sub that displays all of the original buttons that were displayed at the start of the User Control. It then changes the back to Edit page button to the Back to Admin main menu button. Finally, it moves the title to the new location. This is due to the fact that the title is moved to a different position when editing Maths & Physics questions therefore, it moves the title back in position when the button is clicked.

```

Private Sub MQtnbtn_Click(sender As Object, e As EventArgs) Handles MQtnbtn.Click
    'Sub that loads the Maths Question List View, the text boxes and its buttons when the
    'Edit Maths Questions button is clicked

    MQtnShow() 'Sub that loads the Maths User List View and its delete button
    BtnsHide() 'Sub that hides the menu buttons
    EditBtn.Visible = True 'Edit button is visible
    BackBtn.Visible = False 'Back button is hidden
    TitleLbl.Location = New Point(15, 35) 'Moves the title to a new position
End Sub

```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

When the Edit Maths Questions button is clicked it loads this sub. It moves the title to a new position in order to accommodate for the 3 new buttons that are used to edit questions within the list view. It also hides the back to Admin main menu button and displays the back to Edit page button

```
Private Sub MQtnShow() 'Sub that loads the Maths Question List View, the text boxes and its buttons
    MQtnLst.Visible = True
    Qtntxt.Visible = True
    A1txt.Visible = True
    A2txt.Visible = True
    A3txt.Visible = True
    A4txt.Visible = True
    QtnLbl.Visible = True
    A1Lbl.Visible = True
    A2Lbl.Visible = True
    A3Lbl.Visible = True
    A4Lbl.Visible = True
    MRefreshBtn.Visible = True
    MQtnAddBtn.Visible = True
    MQtnDeleteBtn.Visible = True
    MQuestionsList() 'Sub that fills the List View with the Maths Questions
End Sub
```

```
Private Sub BtnsHide() 'Sub that hides the menu buttons
    UsrLstBtn.Visible = False
    MQtnbtn.Visible = False
    PQtnbtn.Visible = False
End Sub
```

```
Private Sub MQuestionsList() 'Sub that loads all the questions and answers from the MQuestions table into the List View
    Dim dt As New DataTable 'Copies a table within the database into vb to be edited and then updated
    dt = runSQL("SELECT * FROM MQuestions ORDER BY MQuestion ASC;")
    'Loads everything from the MQuestions table in ascending order

    For x = 0 To dt.Rows.Count - 1
        Dim items(4) As String '1 dimensional array with 5 values
        items(0) = dt.Rows(x)(0) 'Question
        items(1) = dt.Rows(x)(1) 'Correct Answer
        items(2) = dt.Rows(x)(2) 'Answer 2
        items(3) = dt.Rows(x)(3) 'Answer 3
        items(4) = dt.Rows(x)(4) 'Answer 4

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items array into the LineNew
        MQtnLst.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub
```

Loads Maths Questions list view and its 3 buttons that go with it. Loads the labels and textboxes that will be used to edit questions. Hides all of the original menu buttons at the start of the UC. And fills the Maths Questions list view with all the items in the Maths Questions table within the database.

Question	Correct Ans	Wrong Ans	Wrong Ans	Wrong Ans
Another Blank Question	Another A1	Another A2	Another A3	Another A4
Blank Question	Answer 1	Answer 2	Answer 3	Answer 4
Find $\frac{dy}{dx}$ of $y = 3x^2 + x$	$\frac{dy}{dx} = 6x \dots$	$\frac{dy}{dx} = 6x$	$\frac{dy}{dx} = 3\dots$	$\frac{dy}{dx} = 6x\dots$
Find $\frac{dy}{dx}$ of $y = 6x^2 + 2x$	$\frac{dy}{dx} = 12x\dots$	$\frac{dy}{dx} = 12x$	$\frac{dy}{dx} = 3\dots$	$\frac{dy}{dx} = 6x\dots$
Simplify $15ax^2/5x$	$3ax$	$3ax^2$	$5ax^2$	$5ax$
What is used to show differentiation of the ...	$dy/dx$	$dx/dy$	$y/x$	$dy/df$

Question:  Correct Ans:

Wrong Ans:  Wrong Ans:  Wrong Ans:

```

Private Sub MQtnLst_SelectedIndexChanged(sender As Object, e As EventArgs) Handles MQtnLst.Click
    'Sub that fills the textboxes with the row clicked in the Maths List View

    Qtntxt.Text = MQtnLst.SelectedItems(0).SubItems(0).Text 'Fills textbox with the Question
    A1txt.Text = MQtnLst.SelectedItems(0).SubItems(1).Text 'Fills textbox with the Right Answer
    A2txt.Text = MQtnLst.SelectedItems(0).SubItems(2).Text 'Fills textbox with Answer 2
    A3txt.Text = MQtnLst.SelectedItems(0).SubItems(3).Text 'Fills textbox with Answer 3
    A4txt.Text = MQtnLst.SelectedItems(0).SubItems(4).Text 'Fills textbox with Answer 4

End Sub

```

Whichever row is selected in the list view the text boxes are filled with the corresponding items.

Question:

Correct Ans:

Wrong Ans:

Then there are the 3 buttons (delete, add and update).

```

Private Sub MRefreshBtn_Click(sender As Object, e As EventArgs) Handles MRefreshBtn.Click
    'This sub updates the Maths question by deleting it from the Maths Question table then adding it again

    runSQL("DELETE * FROM MQuestions WHERE MQuestion = '" & MQtnLst.SelectedItems(0).SubItems(0).Text & "'")
    runSQL("INSERT into MQuestions (MQuestion, MA1, MA2, MA3, MA4) values ('" & Qtntxt.Text & "','" &
        Altxt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" & A4txt.Text & "')")
    'Deletes the whole row in the Maths Question table that contains the question selected
    'Adds the question and the 4 answers within the textboxes into the same table

    'Updates the listview
    MQtnLst.Items.Clear() 'Deletes all the items from the listview
    MQuestionsList() 'Runs the sub routine to load the new MQuestion table
    MsgBox("Question Edited!") 'Message box informing the user that the question has been updated

End Sub

```

This sub runs when the Update button is clicked. It deletes whichever row has been selected from the Maths Questions table and then it inserts all the text within the textboxes into each field of the Maths Questions table, therefore updating the question. The sub then clears the table and then calls upon MQuestionList sub that fills the table again, the message box just informs the user that the question has been added. The reason I cleared the items in the list view first is due to the fact that the sub that fills the list view would just add to it therefore creating duplicate values for every question apart from the one just added.

Before developing my other buttons, I decided to test this button just to make sure it was working properly. Thus, this button has the components of both deleting and adding so if this button works fine so should the other 2. One issue I kept having was the fact if no row was selected and I clicked Update the program would crash.

```

Private Sub MRefreshBtn_Click(sender As Object, e As EventArgs) Handles MRefreshBtn.Click
    'This sub updates the Maths question by deleting it from the Maths Question table then adding it again

    runSQL("DELETE * FROM MQuestions WHERE MQuestion = '" & MQtnLst.SelectedItems(0).SubItems(0).Text & "'")
    runSQL("INSERT into MQuestions (MQuestion, MA1, MA2, MA3, MA4) values ('" & Qtntxt.Text & "','" &
Exception Unhandled
    A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" & A4txt.Text & "')")
    'Deletes the whole row in the Maths Question table that contains the question selected
    'Adds the question and the 4 answers within the textboxes into the same table

    'Updates the listview
    MQtnLst.Items.Clear() 'Deletes all the items from the listview
    MQtnList() 'Runs the sub routine to load the new MQuestion table
    MsgBox("Question Edited!") 'Message box informing the user that the question has been updated

    Catch ex As Exception
        'The try statement allows the program not to crash when an item in the listview is not selected
        'It works by the program trying to perform the the sql instructions but if it can't then it just makes the button do nothing
    End Try
End Sub

```

This invalid type of data broke the program since it cannot delete nothing from the table so, I added the following If statement into the sub.

```

If Qtntxt.Text = "" Then
    Exit Sub
    'Stops the program from breaking when the update button is clicked without any row selected
End If

```

So now if the question text box is empty then the SQL statement will not run, however this still never completely fixed the problem. If a user had clicked a row in the list view that then filled the textboxes, then they had clicked off a row within the list view and next decided to click the Update button the program would break again, and the same error message would appear again. Also, if the user had erased all the text from the other text boxes and then clicked the Update button after not having a row selected it would also break the program again. Thus, making the If statement a failure since it could not stop my code from breaking, so I deleted it.

```

Private Sub MRefreshBtn_Click(sender As Object, e As EventArgs) Handles MRefreshBtn.Click
    'This sub updates the Maths question by deleting it from the Maths Question table then adding it again

    Try
        runSQL("DELETE * FROM MQuestions WHERE MQuestion = '" & MQtnLst.SelectedItems(0).SubItems(0).Text & "'")
        runSQL("INSERT into MQuestions (MQuestion, MA1, MA2, MA3, MA4) values ('" & Qtntxt.Text & "','" &
A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" & A4txt.Text & "')")
        'Deletes the whole row in the Maths Question table that contains the question selected
        'Adds the question and the 4 answers within the textboxes into the same table

        'Updates the listview
        MQtnLst.Items.Clear() 'Deletes all the items from the listview
        MQtnList() 'Runs the sub routine to load the new MQuestion table
        MsgBox("Question Edited!") 'Message box informing the user that the question has been updated

    Catch ex As Exception
        'The try statement allows the program not to crash when an item in the listview is not selected
        'It works by the program trying to perform the the sql instructions but if it can't then it just makes the button do nothing
    End Try
End Sub

```

I finally was able to fix the error by placing the whole contents of the sub into a try statement, this meant that if the SQL statement could not delete ‘nothing’ from the table then the button will just be made redundant. And now no matter if the row is not selected or if there is no text in any of the textboxes the button will not delete it and add it or crash the program. This also has created a form of validation since all of the text boxes has to have some text in it for the

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

button to work. So now with all of my subs that involve subs that have to delete from tables I placed all their contents into a try statement so that they do not break the program when now row is selected.

```
Private Sub MQtnAddBtn_Click(sender As Object, e As EventArgs) Handles MQtnAddBtn.Click
    'Sub that adds the Maths Question to the Maths Questions table when they click the Add button

    Try
        runSQL("INSERT into MQuestions (MQuestion, MA1, MA2, MA3, MA4) values ('" & Qtntxt.Text & "','" &
            A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" & A4txt.Text & "')")
        'Adds the question and the 4 answers within the textboxes into the Maths Question table

        'Updates the listview
        MQtnLst.Items.Clear() 'Deletes all the items from the listview
        MQuestionslist() 'Runs the sub routine to load the new MQuestion table
        MsgBox("Question Added!") 'Message box informing the user that the question has been added

    Catch ex As Exception
        'The try statement allows the program not to crash when an item in the listview is not selected
        'It works by the program trying to perform the the sql instructions but if it can't then it just makes the button do nothing
    End Try
End Sub
```

This sub runs when the Add button is selected and as with the Update sub it just inserts the text from the text boxes into the Maths Question table, clears the list view, fills the list view with the new table and then informs the user that the question has been added.

```
Private Sub MQtnDeleteBtn_Click(sender As Object, e As EventArgs) Handles MQtnDeleteBtn.Click
    'Sub that deletes the Maths question from the Maths Question Table

    Dim DeletedCheck As Boolean = False 'Creates the false boolean value that will be used to with deleting the user

    Try
        If MsgBox("Are you sure you want to delete this question? ", MsgBoxStyle.YesNo, "Delete Question") = MsgBoxResult.Yes Then
            runSQL("DELETE * FROM MQuestions WHERE MQuestion = '" & MQtnLst.SelectedItems(0).SubItems(0).Text & "'")
            'If the user chooses the yes option in the message box then
            'It deletes the whole row from the Maths Question table that contains the question selected in the list view
            DeletedCheck = True 'Turns the boolean variable into true

            'Updates the listview
            MQtnLst.Items.Clear() 'Deletes all the items from the listview
            MQuestionslist() 'Runs the sub routine to load the new MQuestion table
        End If
        'If they choose No then the question would not be deleted and the message box will close

        If DeletedCheck = True Then
            MsgBox("Question Deleted!")
        End If
        'If the user choose yes in the user box it shows this message box afterwards informing the user that the question has been deleted

    Catch ex As Exception
        'The try statement allows the program not to crash when an item in the listview is not selected
        'It works by the program trying to perform the the sql instructions but if it can't then it just makes the button do nothing
    End Try
End Sub
```

Sub runs when Delete button is clicked. When it is clicked a message box appears to make verify whether the user is certain they wish to delete the question from the Maths Questions table. If yes, then it deletes the question from the table and then the Boolean variable is assigned to true. When this variable is true another message box appears informing the user that the question has been deleted. If the user had clicked no, then the question will not be deleted and thus the Deleted Check variable would still be false, so no message box would load.

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

**\*All the above was for when the Maths Question Edit button is clicked, when the Physics Question Edit button is clicked it does the exact same thing, has the exact same subs but it works with the Physics Questions table instead of the Maths Questions table.**

```
Private Sub PQtnbtn_Click(sender As Object, e As EventArgs) Handles PQtnbtn.Click
    'Sub that does the same as above but with the Physics counterpart
    PQtnShow()
    BtnsHide()
    EditBtn.Visible = True
    BackBtn.Visible = False
    TitleLbl.Location = New Point(15, 35)
End Sub
```

```
Private Sub BtnsHide() 'Sub that hides the menu buttons
    UsrLstBtn.Visible = False
    MQtnbtn.Visible = False
    PQtnbtn.Visible = False
End Sub
```

```
Private Sub PQtnShow() 'Sub that loads the Physics Question List View, the text boxes and its buttons
    PQtnLst.Visible = True
    Qtntxt.Visible = True
    A1txt.Visible = True
    A2txt.Visible = True
    A3txt.Visible = True
    A4txt.Visible = True
    QtnLbl.Visible = True
    A1lbl.Visible = True
    A2lbl.Visible = True
    A3lbl.Visible = True
    A4lbl.Visible = True
    PRefreshBtn.Visible = True
    PQtnAddBtn.Visible = True
    PQtnDeleteBtn.Visible = True
    PQuestionsList() 'Sub that fills the List View with the Physics Questions
End Sub
End Class
```

```
Private Sub PQuestionsList() 'Sub that loads all the questions and answers from the PQuestions table into the List View

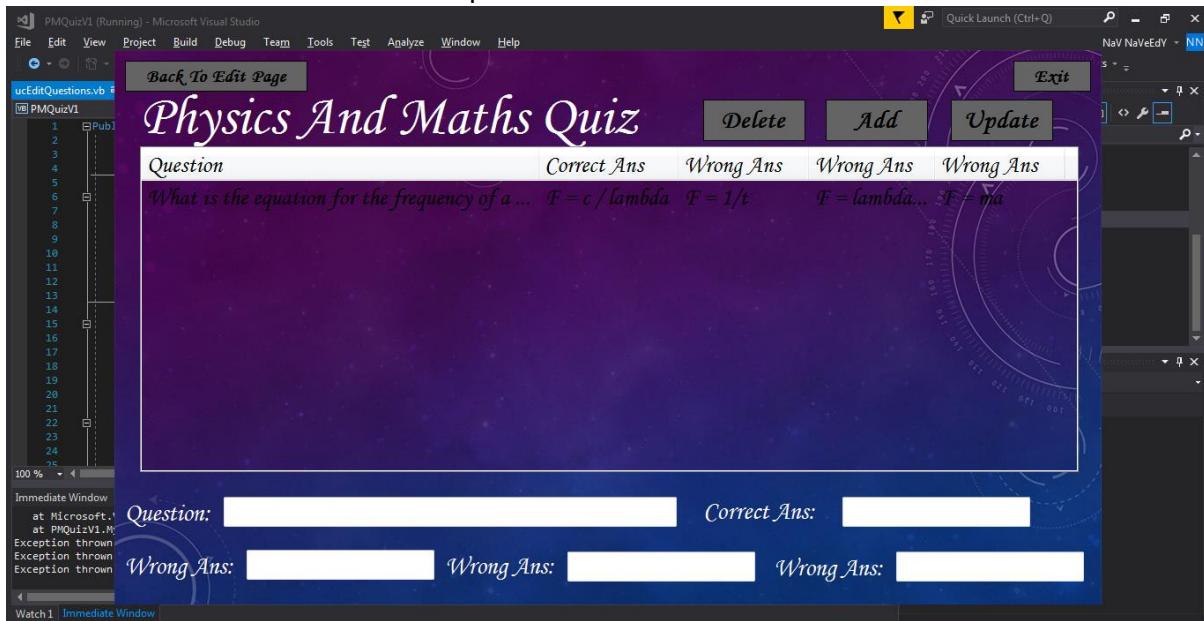
    Dim dt As New DataTable 'Copies a table within the database into vb to be edited and then updated
    dt = runSQL("SELECT * FROM PQuestions ORDER BY PQuestion ASC;")
    'Loads everything from the PQuestions table in ascending order

    For x = 0 To dt.Rows.Count - 1
        Dim items(4) As String '1 dimensional array with 5 values
        items(0) = dt.Rows(x)(0) 'Question
        items(1) = dt.Rows(x)(1) 'Correct Answer
        items(2) = dt.Rows(x)(2) 'Answer 2
        items(3) = dt.Rows(x)(3) 'Answer 3
        items(4) = dt.Rows(x)(4) 'Answer 4

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items array into the LineNew
        PQtnLst.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub
```

Candidate Name: Naveed Ali Rafeeq

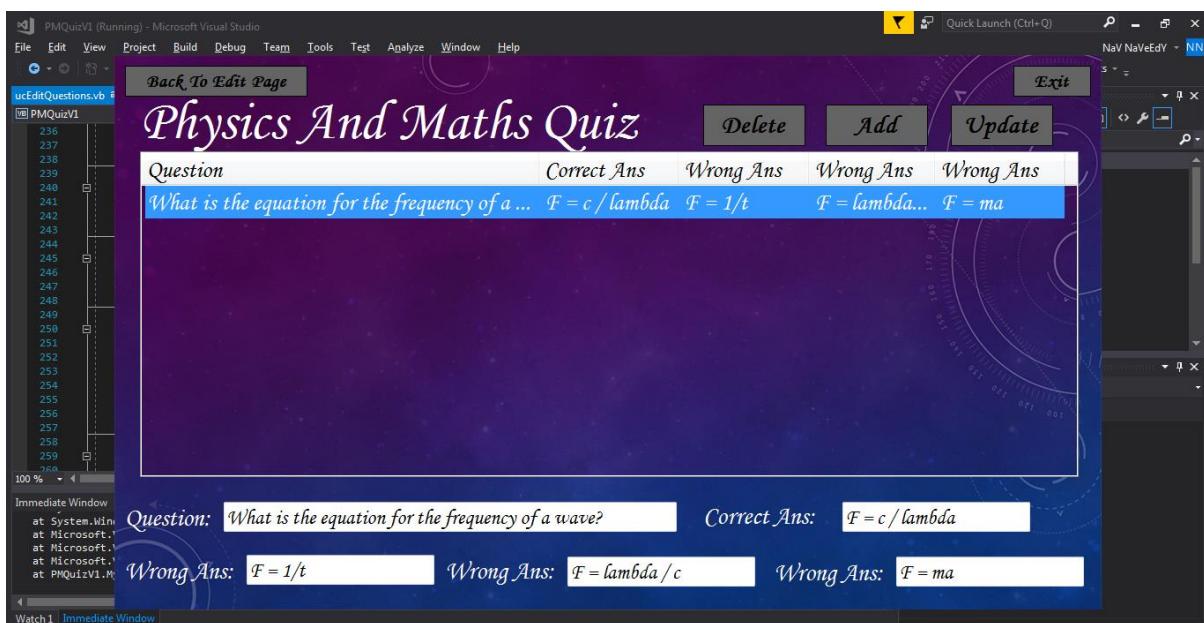
Candidate Number: 1904



```
Private Sub PQtnLst_SelectedIndexChanged(sender As Object, e As EventArgs) Handles PQtnLst.Click
    'Sub that fills the textboxes with the row clicked in the Physics List View

    Qtntxt.Text = PQtnLst.SelectedItems(0).SubItems(0).Text 'Fills textbox with the Question
    A1txt.Text = PQtnLst.SelectedItems(0).SubItems(1).Text 'Fills textbox with the Right Answer
    A2txt.Text = PQtnLst.SelectedItems(0).SubItems(2).Text 'Fills textbox with Answer 2
    A3txt.Text = PQtnLst.SelectedItems(0).SubItems(3).Text 'Fills textbox with Answer 3
    A4txt.Text = PQtnLst.SelectedItems(0).SubItems(4).Text 'Fills textbox with Answer 4

End Sub
```



```

Private Sub PRefreshBtn_Click(sender As Object, e As EventArgs) Handles PRefreshBtn.Click
    'This sub updates the Physics question by deleting it from the Physics Question table then adding it again

    Try
        runSQL("DELETE * FROM PQuestions WHERE PQuestion = '" & PQtnLst.SelectedItems(0).SubItems(0).Text & "'")
        runSQL("INSERT into PQuestions (PQuestion, PA1, PA2, PA3, PA4) values ('" & Qtntxt.Text & "','" &
            A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" & A4txt.Text & "')")
        'Deletes the whole row in the Physics Question table that contains the question selected
        'Adds the question and the 4 answers within the textboxes into the same table

        'Updates the listview
        PQtnLst.Items.Clear() 'Deletes all the items from the listview
        PQuestionsList() 'Runs the sub routine to load the new PQuestion table
        MsgBox("Question Edited!") 'Message box informing the user that the question has been edited

    Catch ex As Exception
        'The try statement allows the program not to crash when an item in the listview is not selected
        'It works by the program trying to perform the the sql instructions but if it can't then it just makes the button do nothing
    End Try
End Sub

```

```

Private Sub PQtnAddBtn_Click(sender As Object, e As EventArgs) Handles PQtnAddBtn.Click
    'Sub does the same as the one above but with the Physics questions table

    Try
        runSQL("INSERT into PQuestions (PQuestion, PA1, PA2, PA3, PA4) values ('" & Qtntxt.Text & "','" &
            A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" & A4txt.Text & "')")

        PQtnLst.Items.Clear()
        PQuestionsList()
        MsgBox("Question Added!")

    Catch ex As Exception
    End Try
End Sub

```

```

Private Sub PQtnDeleteBtn_Click(sender As Object, e As EventArgs) Handles PQtnDeleteBtn.Click
    'Same sub as above but deletes from the Physics Question table
    Dim DeletedCheck As Boolean = False

    Try
        If MsgBox("Are you sure you want to delete this question? ", MsgBoxStyle.YesNo, "Delete Question") = MsgBoxResult.Yes Then
            runSQL("DELETE * FROM PQuestions WHERE PQuestion = '" & PQtnLst.SelectedItems(0).SubItems(0).Text & "'")
            DeletedCheck = True
            PQtnLst.Items.Clear()
            PQuestionsList()
        End If

        If DeletedCheck = True Then
            MsgBox("Question Deleted!")
        End If

    Catch ex As Exception
    End Try
End Sub

```

There are 2 sets of buttons one for Maths and the other for Physics as well as 2 list views however, the text boxes and the labels are all the same. The reason for this is due to the fact that all they do is hold the text values of the items within the list view however the buttons have to read and write to different tables within the database therefore I created 2 of each that look identical.

## Testing Prototype 3

### Navigation:

Test Data	Test Number	Type Of Test Data	Output
LMC on yes button after LMC on the Exit button	1	Valid	Program Closes
RMC on buttons	NA	Invalid	Nothing
LMC not on the buttons	NA	Invalid	Nothing
LMC on any 'Back' button	Passed in previous Prototype	Valid	Disposes current UC and loads another
LMC on no button after LMC on the Exit button	2	Invalid	Message box closes

In the last prototype I did not fully test the Exit button function therefore in this prototype I did. Apart from that all of the basic navigation within the program has already been tested.

#### Admin Menu UC:

Test Data	Test Number	Type Of Test Data	Output
LMC on Leader Boards button	3	Valid	Leader Boards UC loads, and the Admin UC is disposed
RMC on Leader Boards button	NA	Invalid	Nothing
LMC on Edit Questions button	4	Valid	Edit Questions UC loads, and the Admin UC is disposed
RMC on Edit Questions button	NA	Invalid	Nothing
Keyboard	NA	Invalid – Erroneous	Nothing

Like the Main Menu there is only a few buttons that can be tested within the User Control, therefore they are tested within this prototype.

**Leader Boards UC:**

Test Data	Test Number	Type Of Test Data	Output
LMC on username in either Maths or Physics Leader board	5	Valid	Becomes highlighted
RMC on the back ground	NA	Invalid	Nothing
LMC drag on the columns in any Leader board	6	Valid	Columns become resizable
Edit the items in the Leader boards, try this by highlighting a username and trying to edit it by typing	7	Invalid	Nothing

I did the following tests so that they admin is able to make the leaderboards more bespoke to them, but they cannot edit the actual leaderboards itself.

**Edit Questions UC:**

Test Data	Test Number	Type Of Test Data	Output
LMC on username in user list	8	Valid	Row becomes highlighted
RMC row in user list or one of the question lists	NA	Invalid	Nothing
LMC drag on the columns in either User list or question list	9	Valid	Columns become resizable
LMC on any item in a question list	10	Valid	Row becomes highlighted and the textboxes are filled with the items within the selected row
LMC on a Delete or Update button within one of the question list views whilst no row is selected	11	Invalid	Nothing
LMC on a Delete or Update button within one of the question list views whilst one of the textboxes does not have text in it	12	Invalid	Nothing
LMC on Add button	13	Valid	Inserts the text in the textboxes into the corresponding question table and the list view as well
LMC on a Delete button and then	14	Valid	Deletes the Question

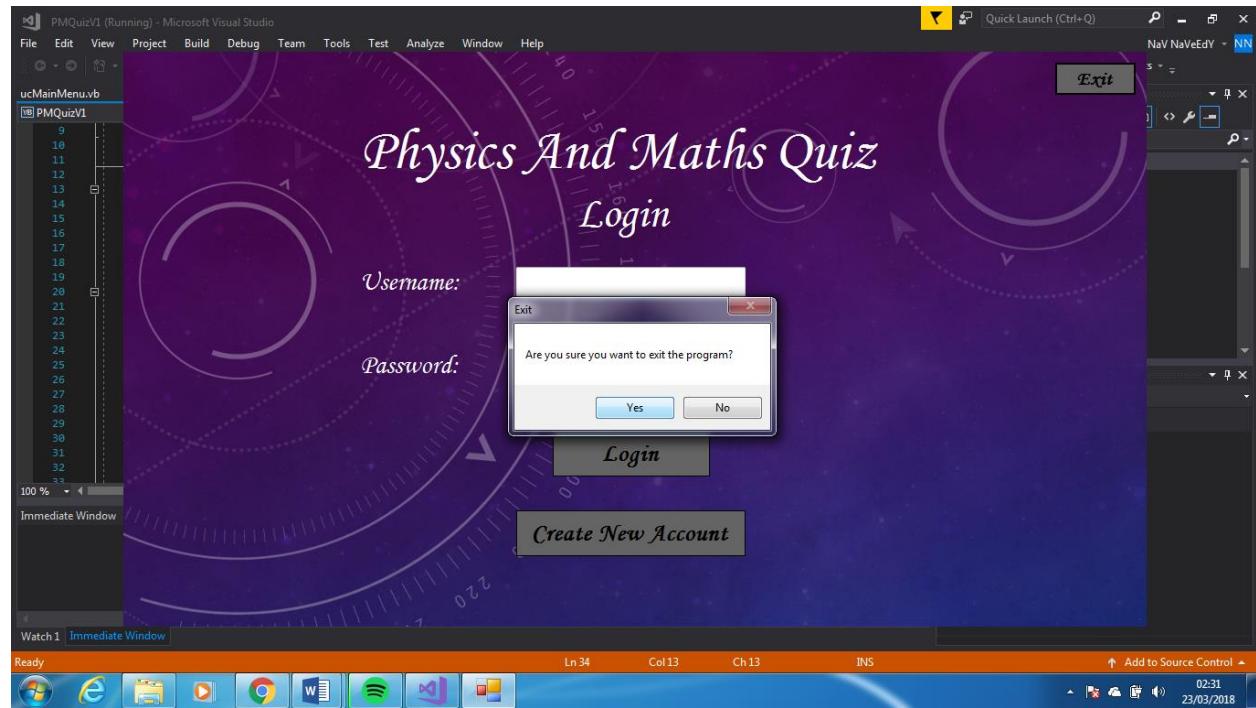
Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

clicking yes on the message box afterwards			from its table and also from the list view
LMC on a Delete button and then clicking no on the message box afterwards	15	Invalid	Nothing happens
LMC on an Update button after editing the text in the text boxes	16	Valid	Makes the changes within the corresponding question table as well as the list view
LMC on an Update button but not changing any of the text from the text boxes	17	Invalid	Message box appears but nothing changes

The tests above reassure that the amendments I made to the validation parts of the edit questions will not break this time. That is why I did many tests without selecting rows etc because I wanted to make sure that the validation would not have any more problems.

### Test 1

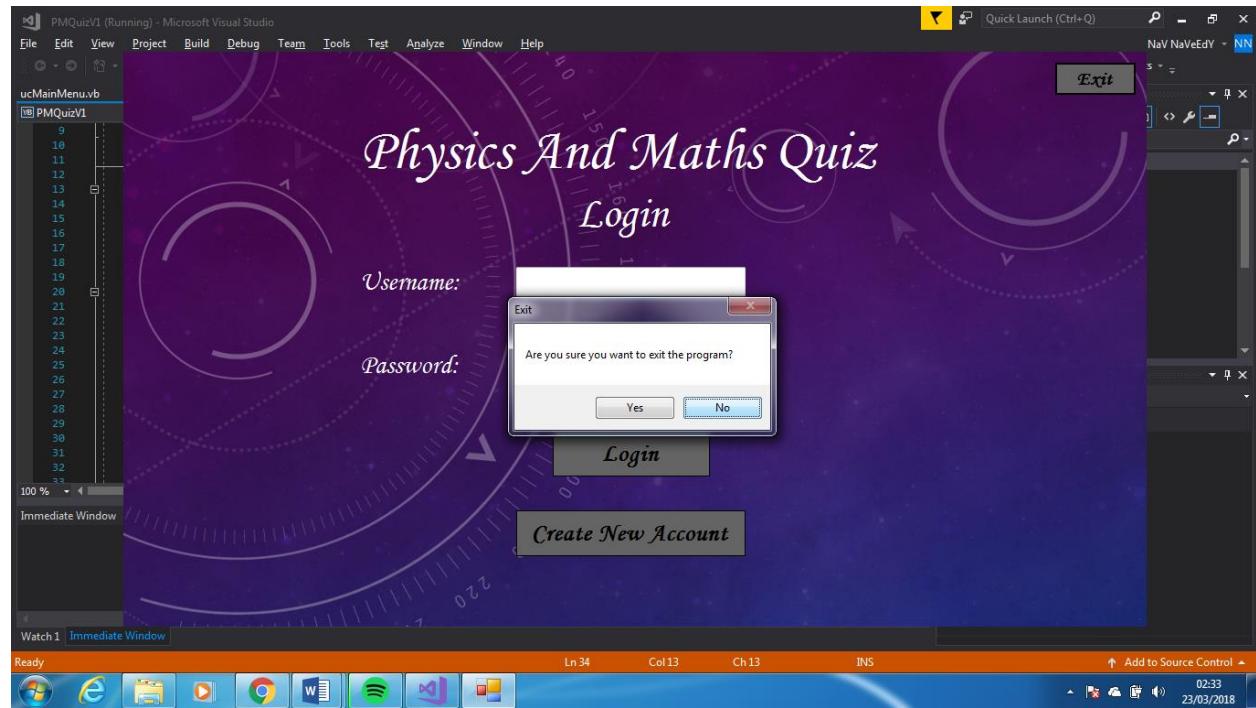


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

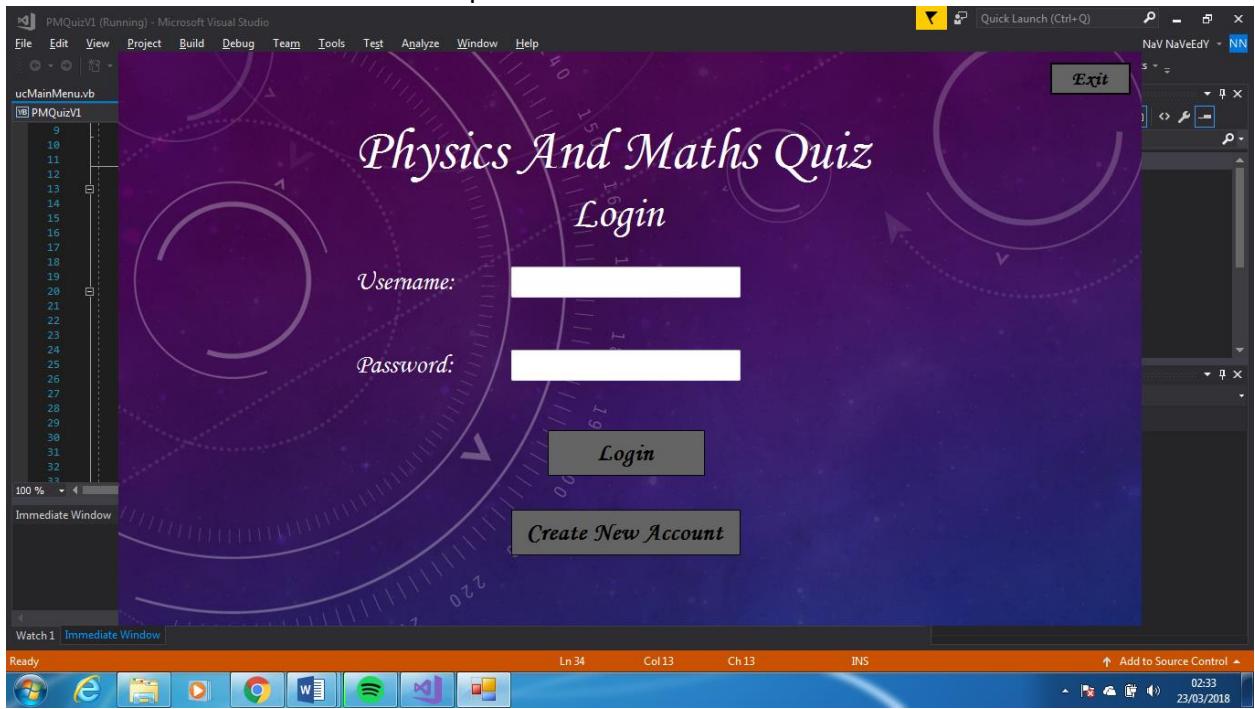
```
9     End If 'When the user presses enter button in the password textbox it runs the login button
10
11 End Sub
12
13 Private Sub LoginBtn_Click(sender As Object, e As EventArgs) Handles LoginBtn.Click
14
15     Dim dtAdmin, dtStudent As New DataTable 'Creates 2 datatable variables
16     dtAdmin = runSQL("Select * From AdminData where TUserR = '" & UsrNameTxt.Text & "' and TPass = '" & PassTxt.Text & "'")
17     dtStudent = runSQL("Select * From StudentData where SUserR = '" & UsrNameTxt.Text & "' and SPass = '" & PassTxt.Text & "'")
18     'Extracts all the Usernames and Passwords from the Admin and Student Tables
19
20     If dtStudent.Rows.Count = 1 Then
21         ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
22         ucLoad.addUC("ucAdminMainmenu") 'Puts this string into the subroutine AddUC which determines which user control to load
23         Me.Dispose() 'Gets rid of the user control
24         GC.Collect() 'Gather all the remaining data left after closing the user control
25         Details.LoginUser = dtStudent.Rows(0)(0) 'Saves the SUserR on the student row to be used later
26         Details.UserID = dtStudent.Rows(0)(2) 'Saves the TID on the student row to be used later
27         Details.UserTID = dtAdmin.Rows(0)(0) 'Saves the TID on the admin row to be used later
28         ElseIf dtAdmin.Rows.Count = 1 Then
29             ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
30             ucLoad.addUC("ucAdminMainmenu") 'Puts this string into the subroutine AddUC which determines which user control to load
31             Me.Dispose() 'Gets rid of the user control
32             GC.Collect() 'Gather all the remaining data left after closing the user control
33             Details.LoginUser = dtAdmin.Rows(0)(1) 'Saves the TUserR on the admin row to be used later
34             Details.UserID = dtAdmin.Rows(0)(0) 'Saves the TID on the admin row to be used later
35         Else
36             MsgBox("Your details have been entered incorrectly, please try again")
37         End If
38
39     End Sub
40
41 Private Sub NewAccBtn_Click(sender As Object, e As EventArgs) Handles NewAccBtn.Click
42     ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to access AddUC
43     ucLoad.addUC("ucCreateAcc") 'Puts this string into the subroutine AddUC which determines which user control to load
44     Me.Dispose() 'Gets rid of the user control
45
46 End Sub
```

## Test 2

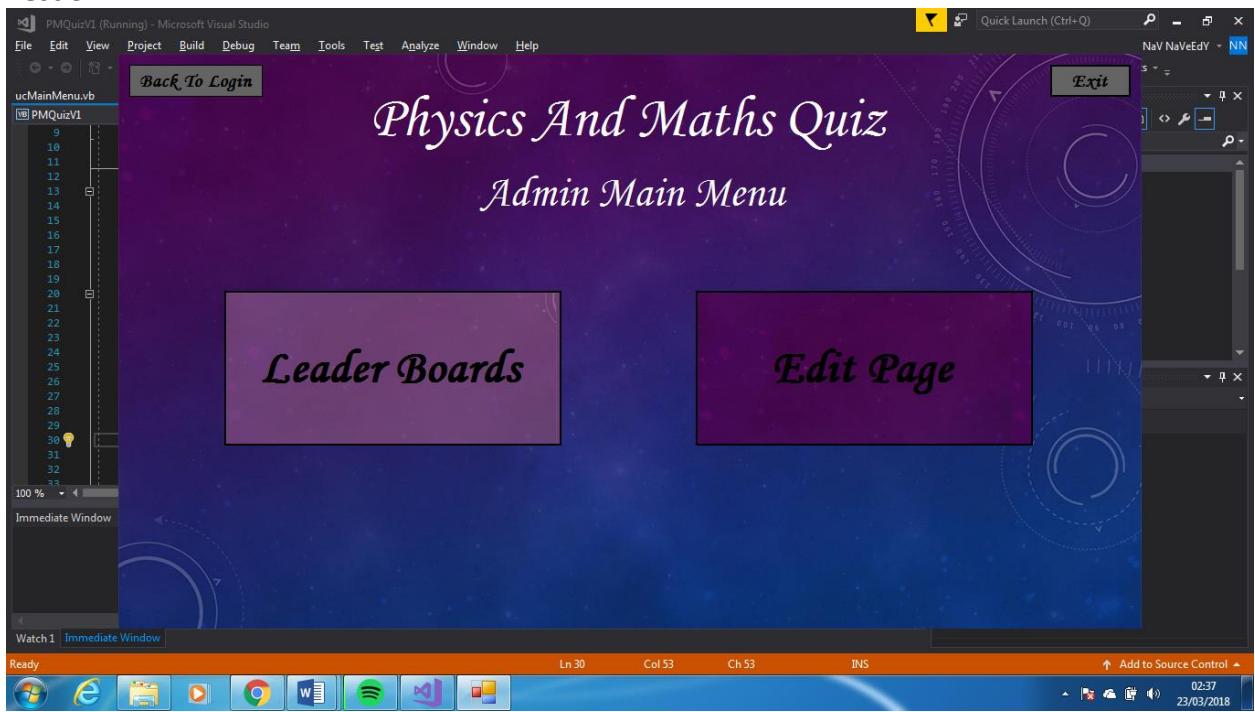


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

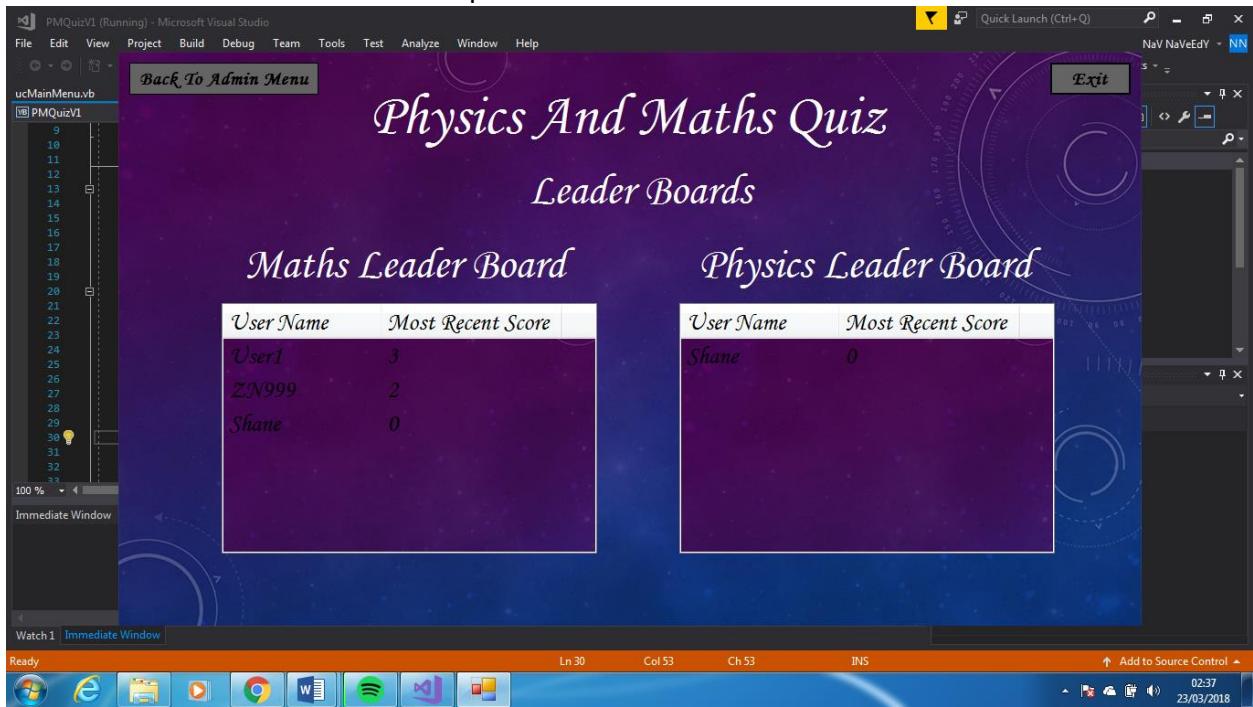


### Test 3

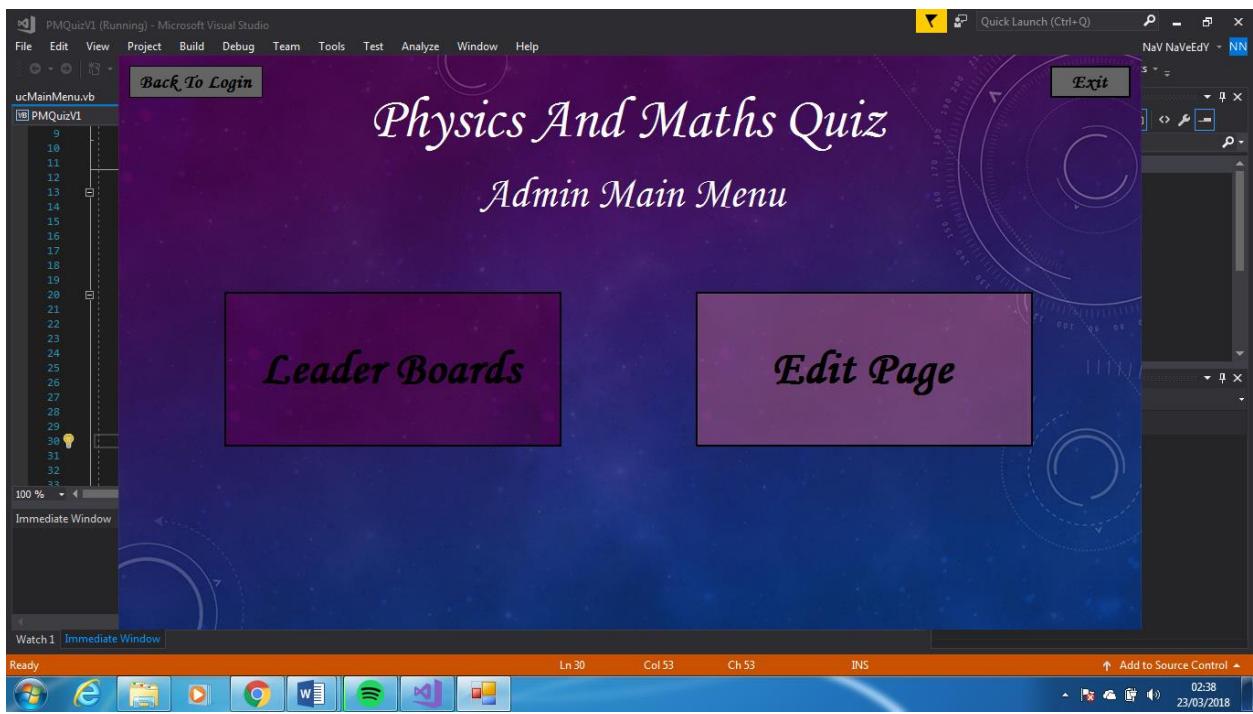


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

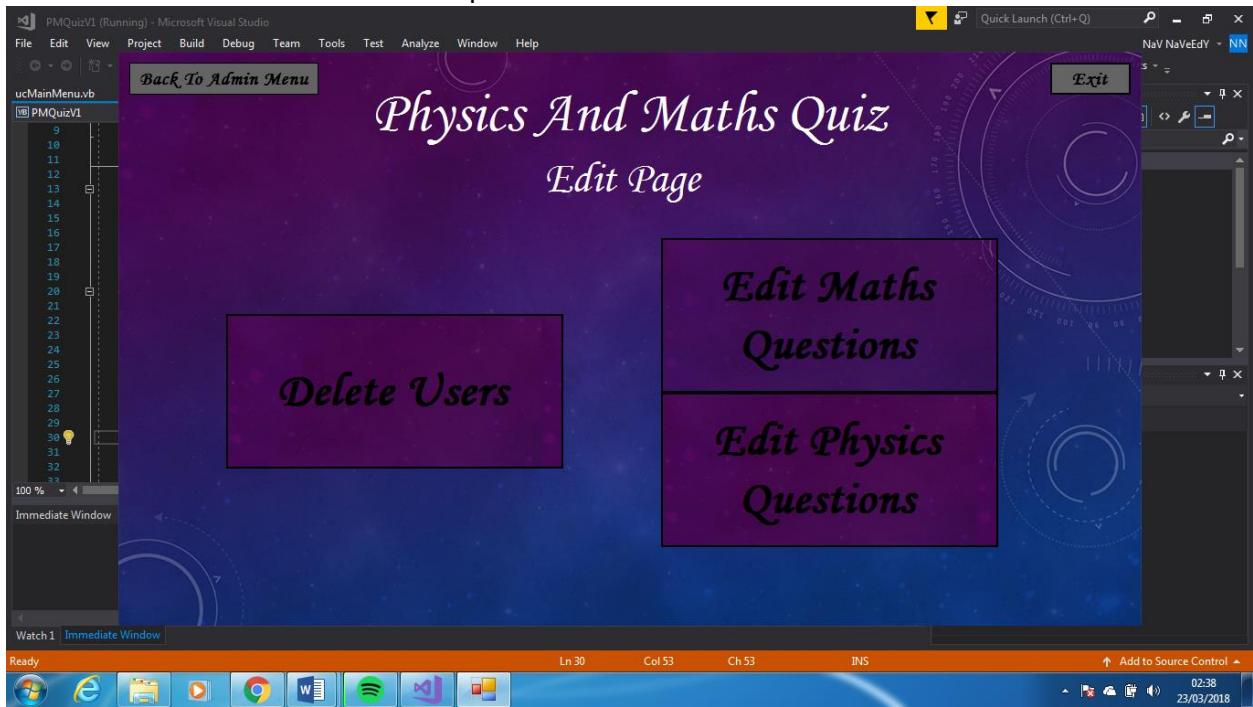


#### Test 4

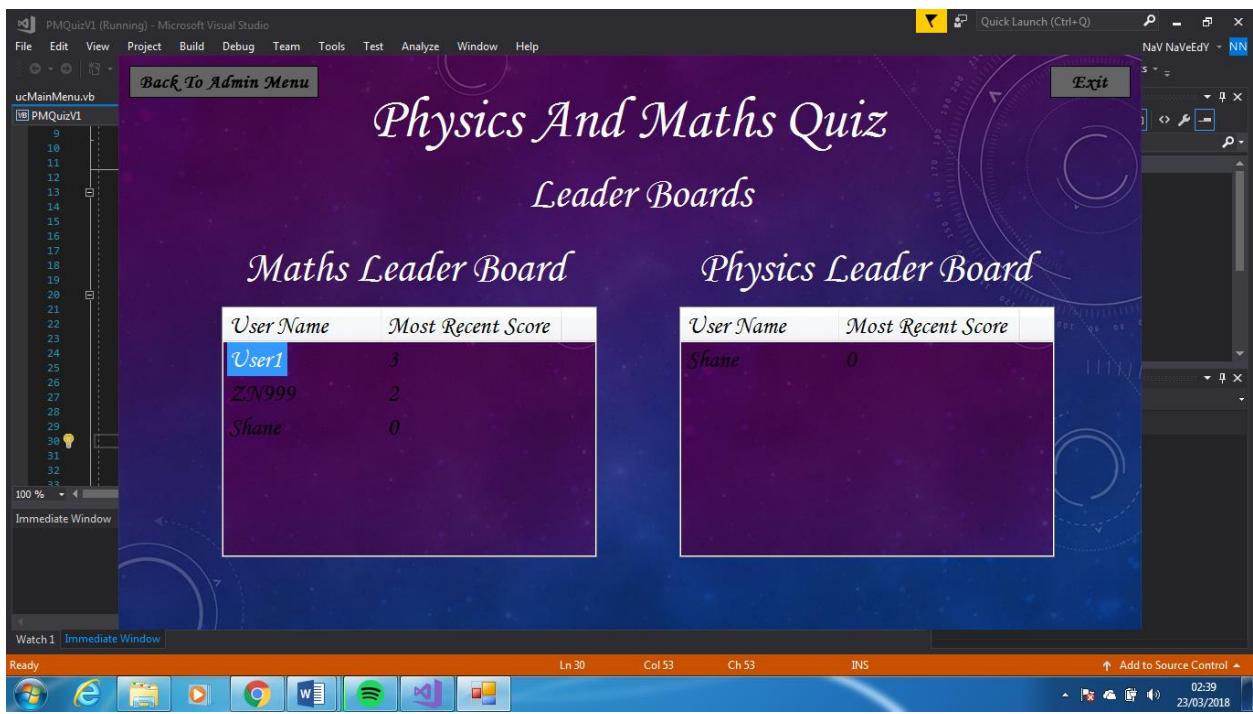


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



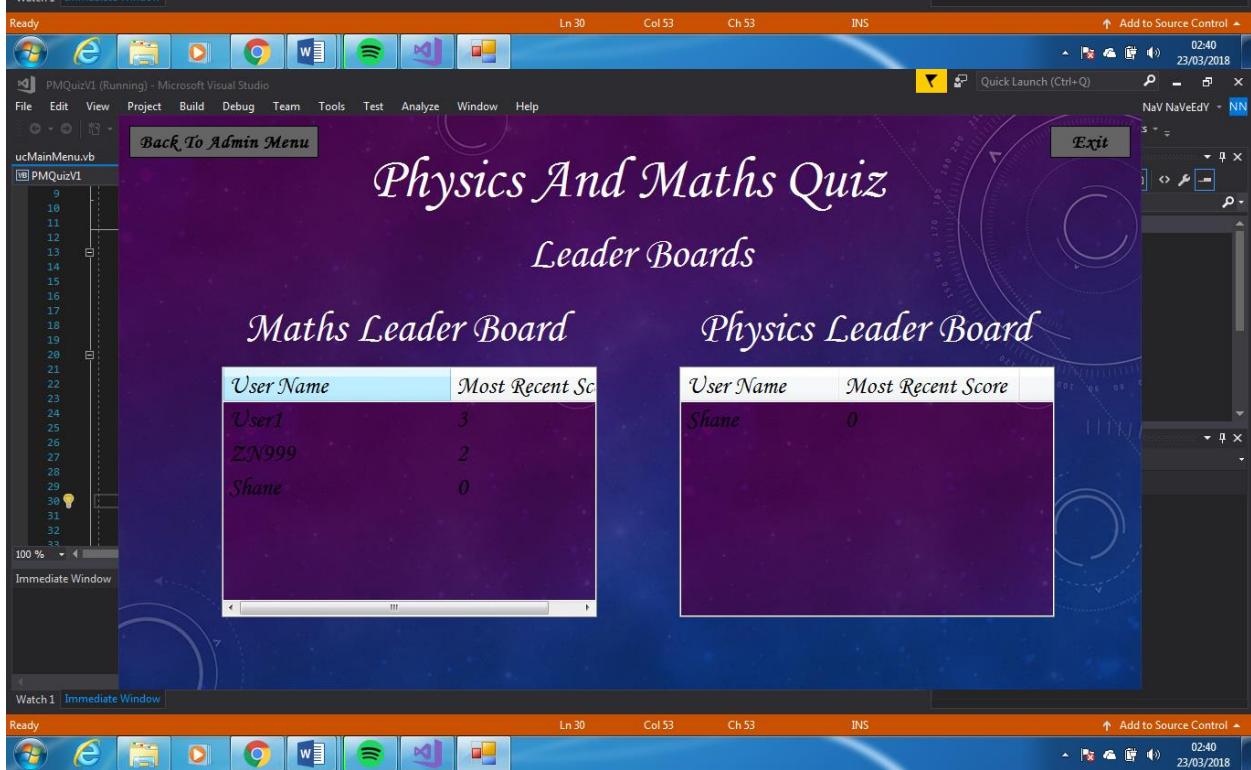
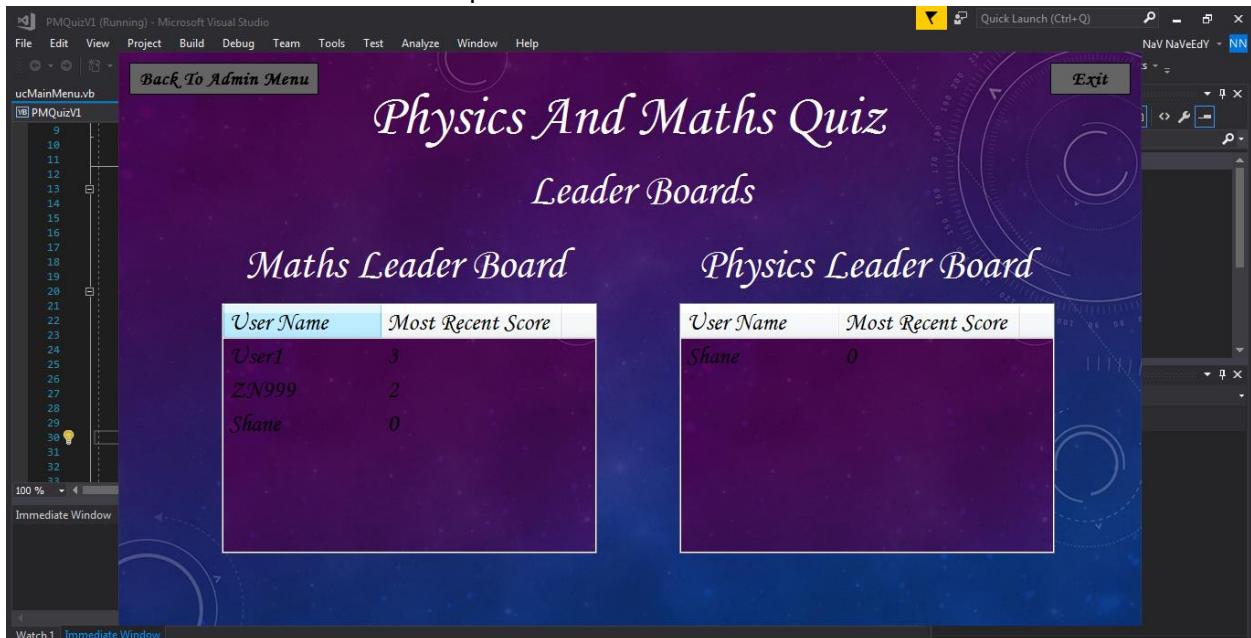
## Test 5



## Test 6

Candidate Name: Naveed Ali Rafeeq

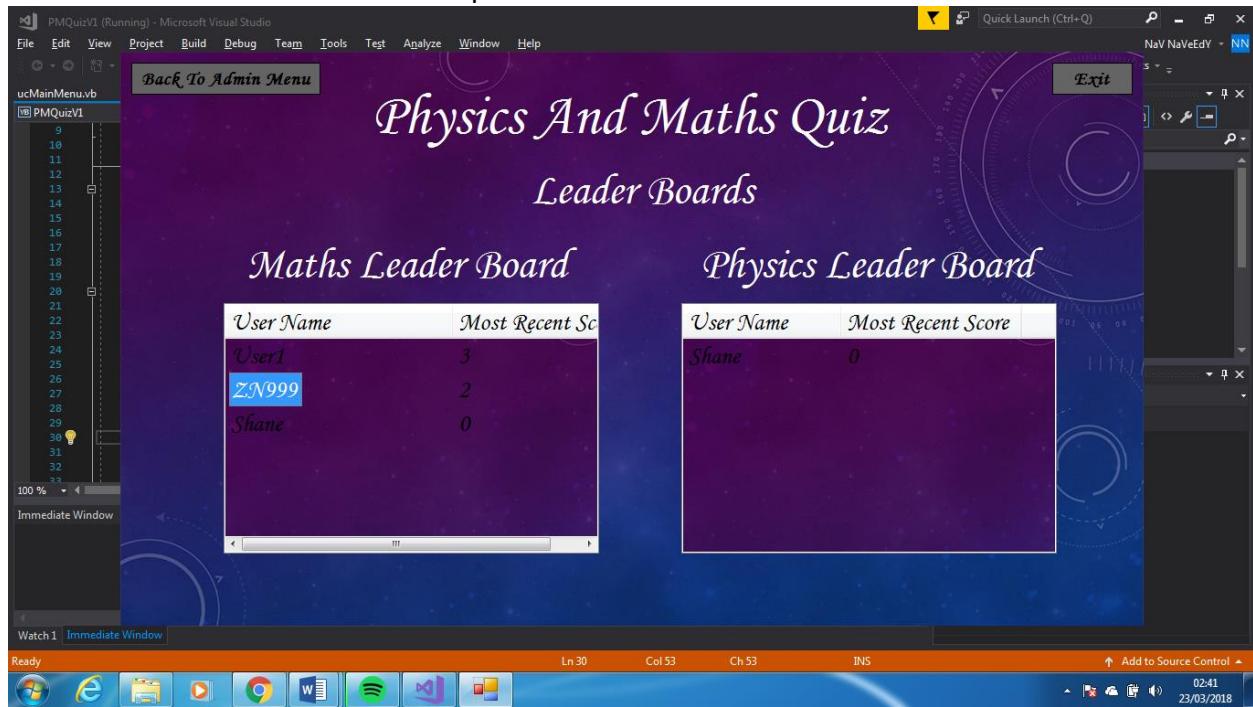
Candidate Number: 1904



## Test 7

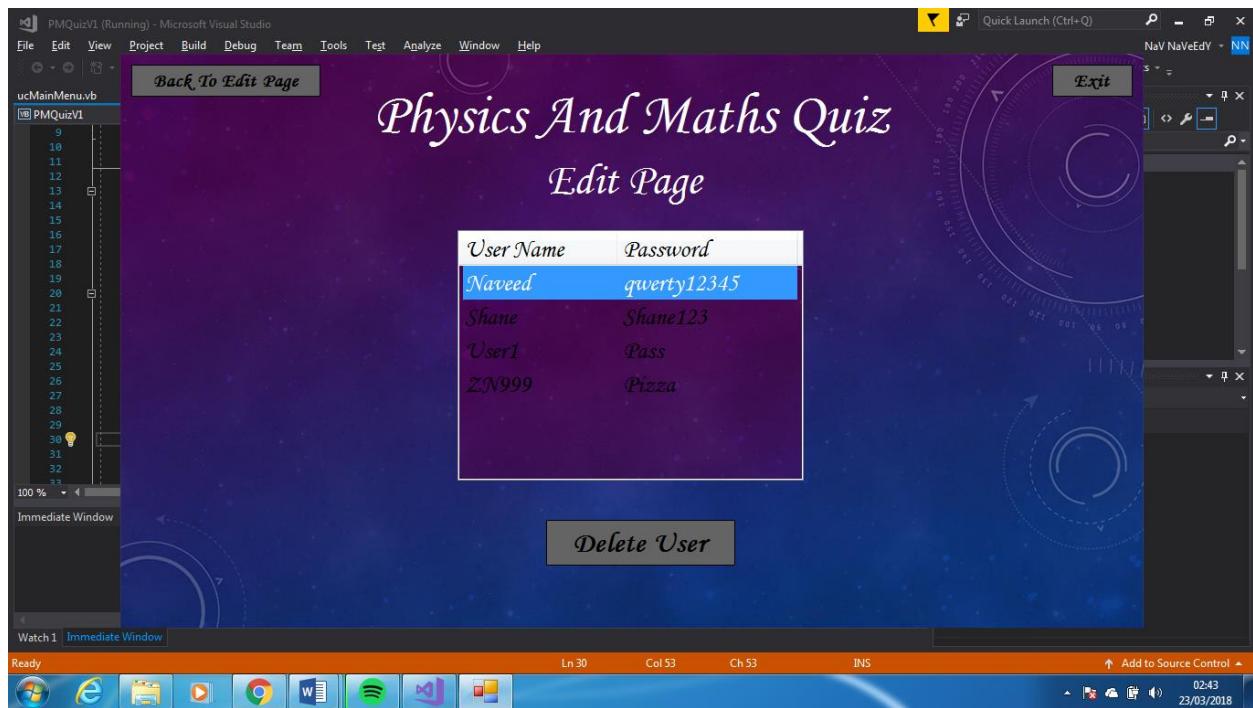
Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



Nothing happens after typing into my keyboard

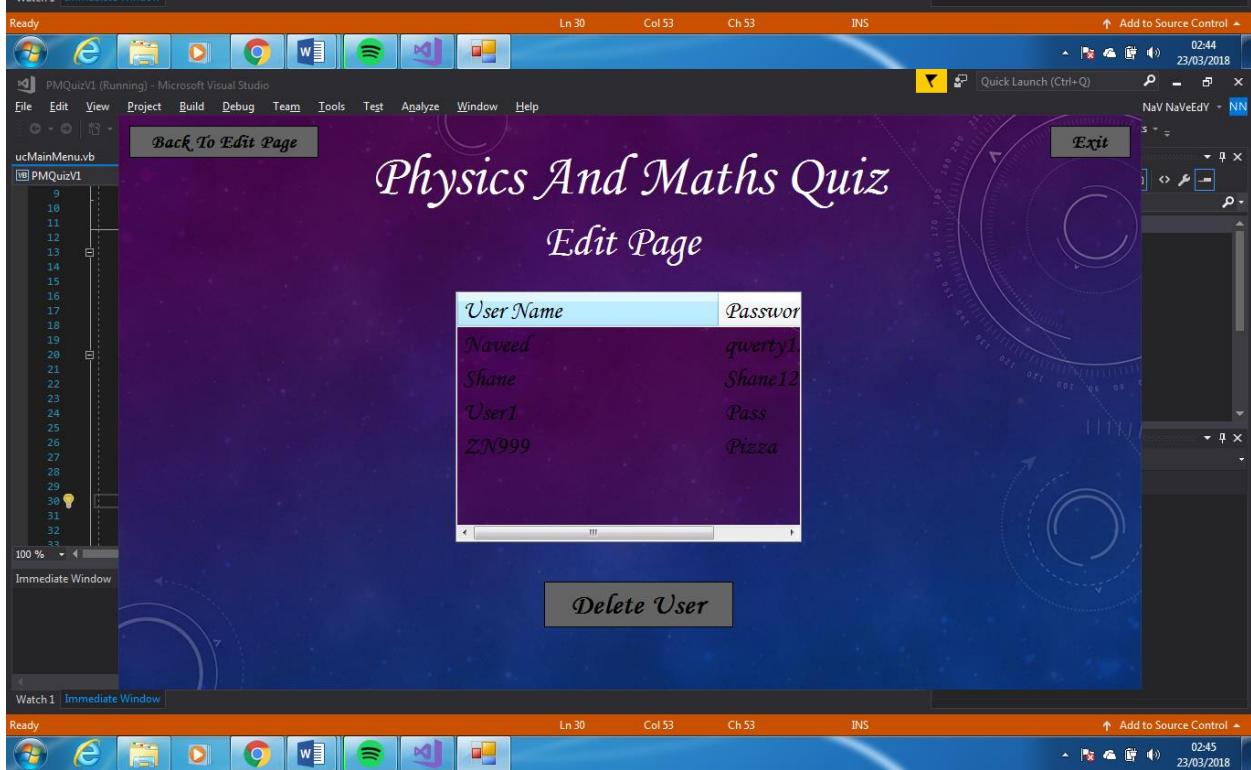
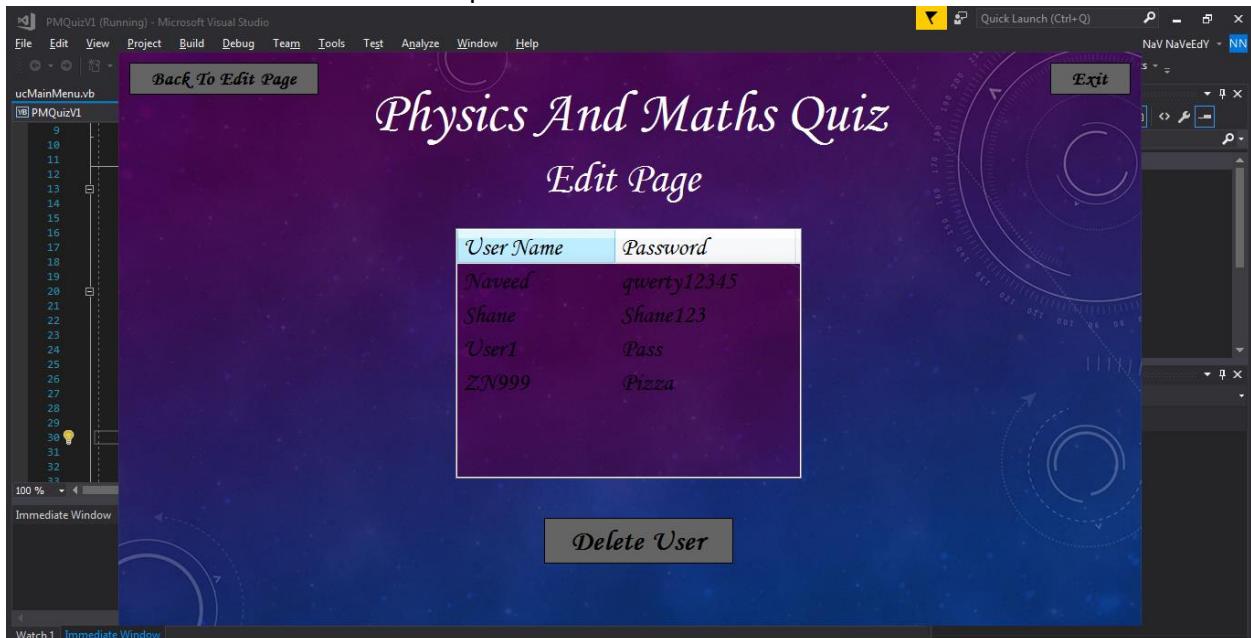
## Test 8



## Test 9

Candidate Name: Naveed Ali Rafeeq

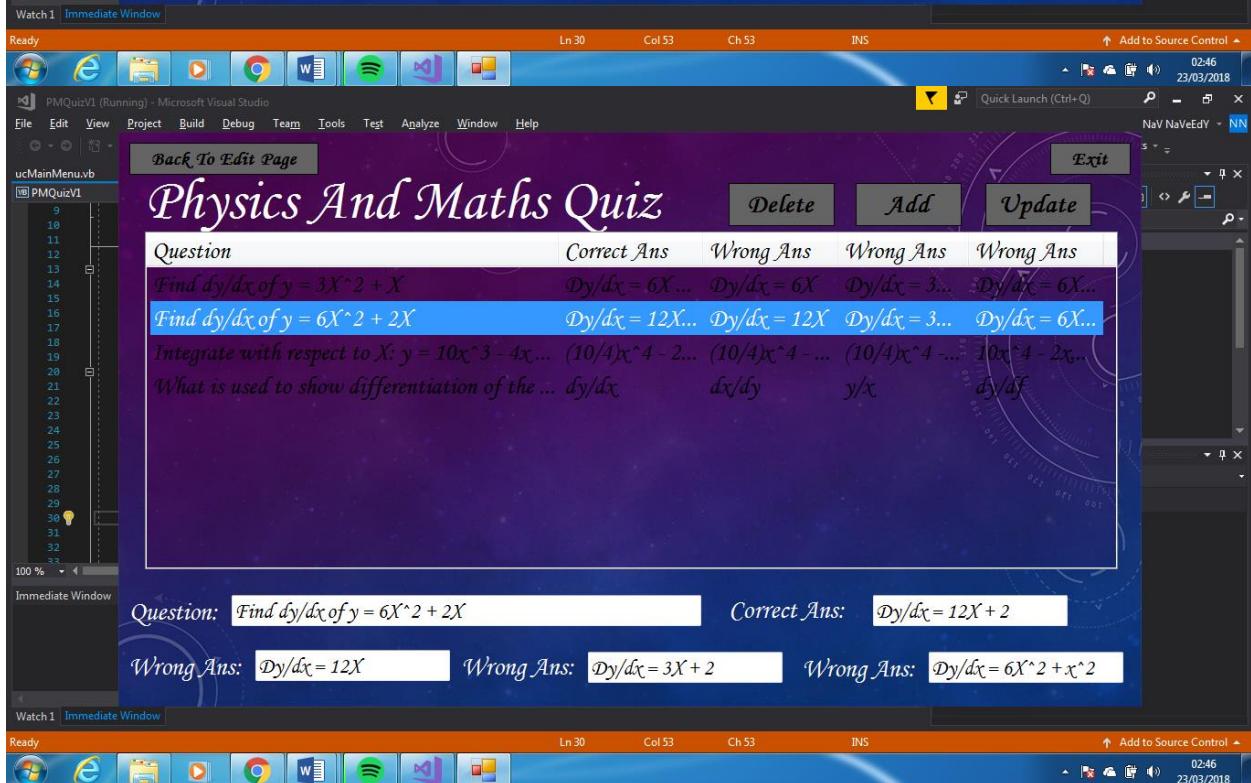
Candidate Number: 1904



## Test 10

Candidate Name: Naveed Ali Rafeeq

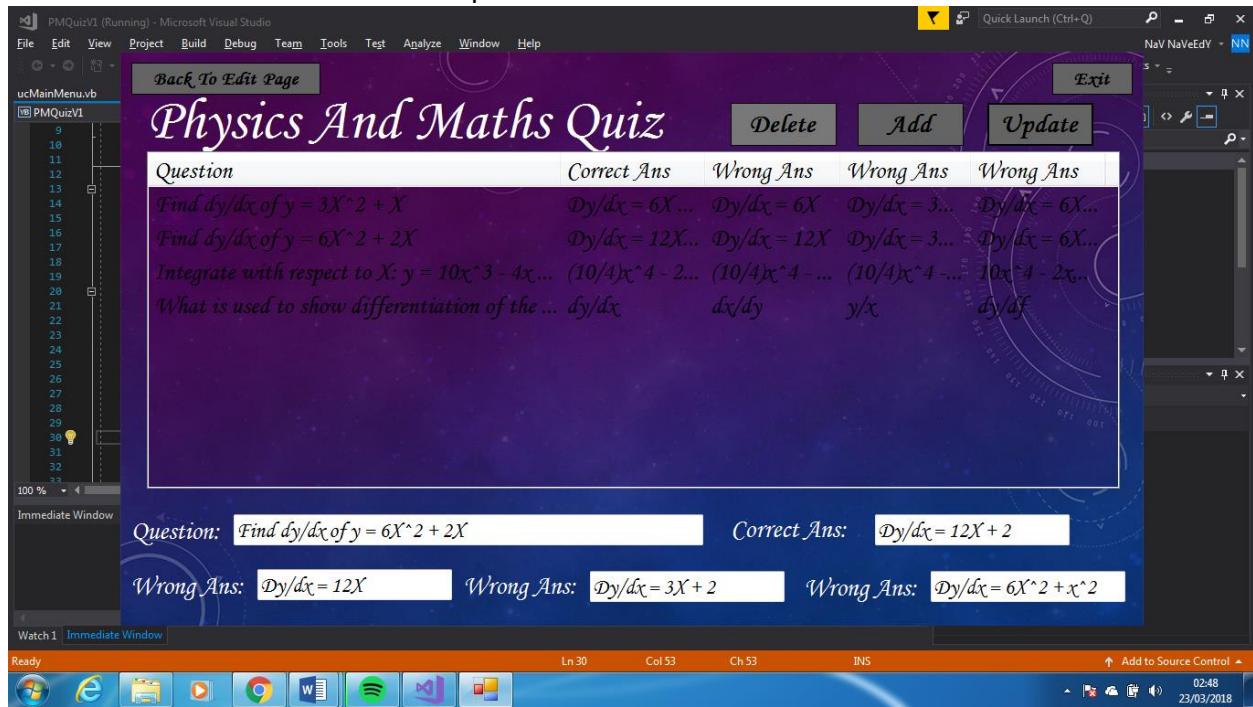
Candidate Number: 1904



## Test 11

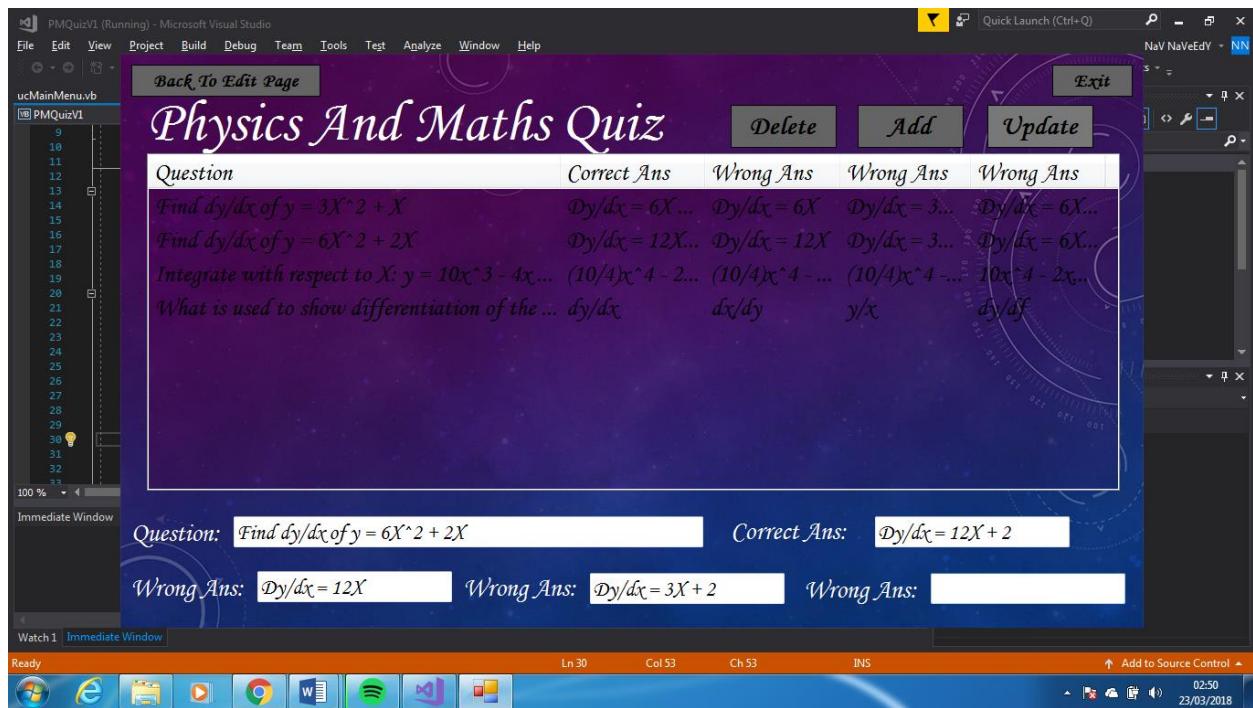
Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



As you can see by the delete button being highlighted I have clicked it yet nothing has happened

## Test 12

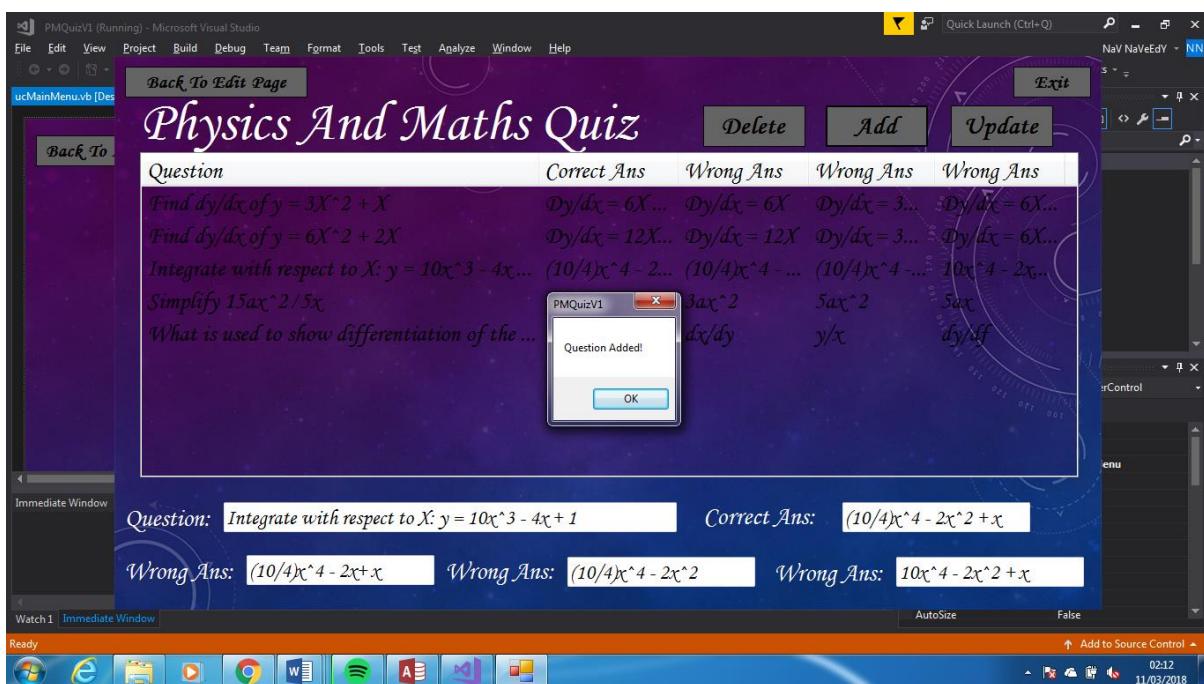
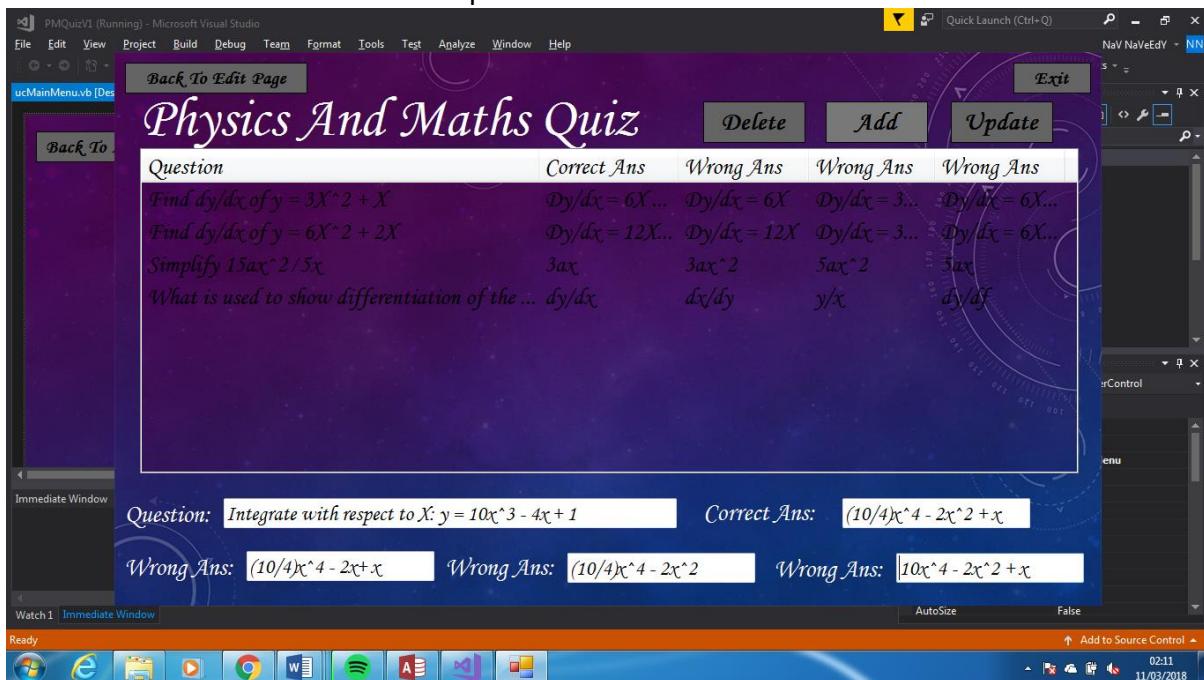


Same as above but with the Update button

## Test 13

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

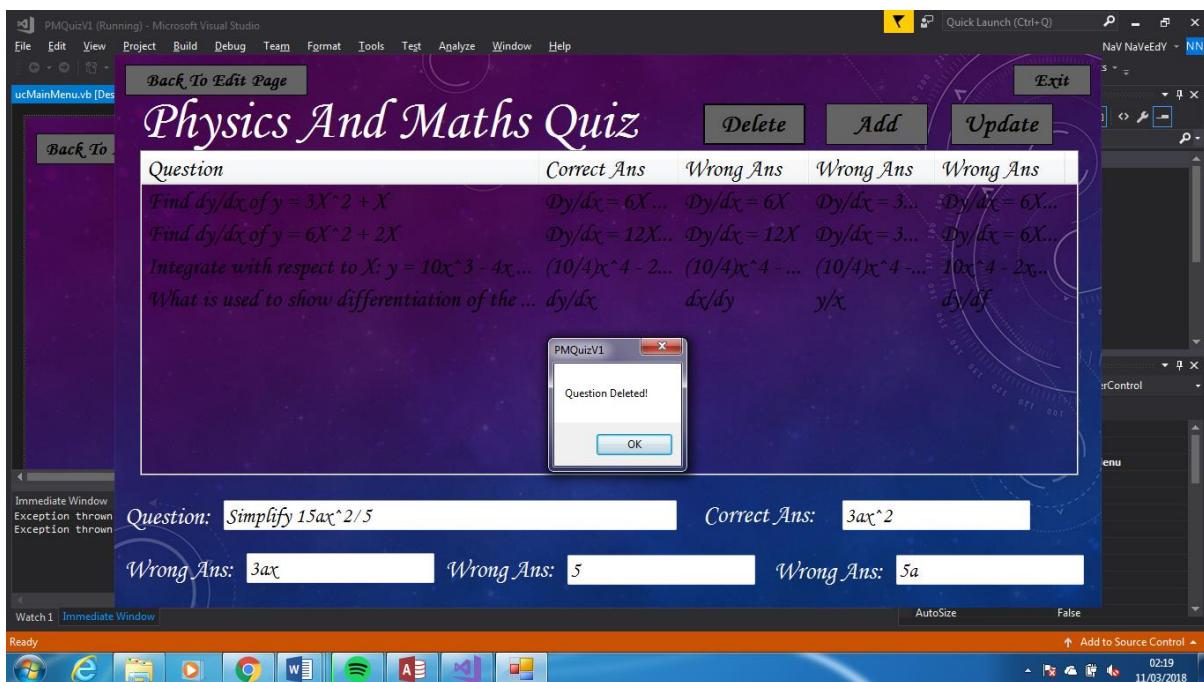
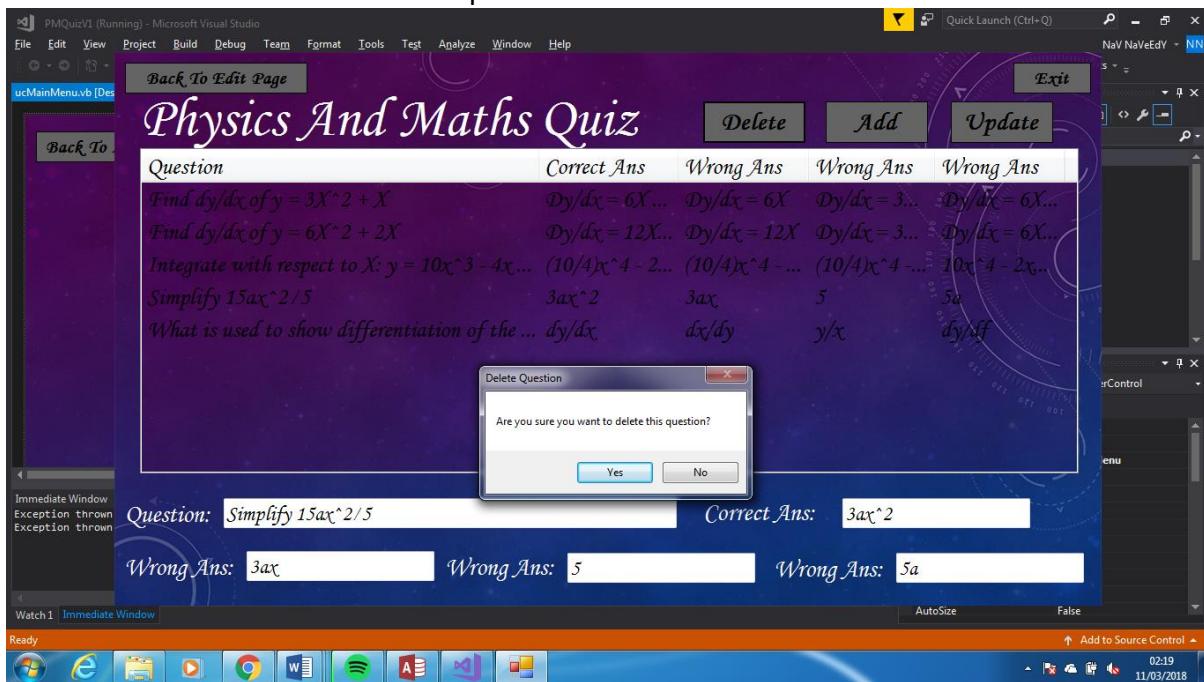


MQuestion	MA1	MA2	MA3	MA4	Click to Add
Find dy/dx of y = 3X^2 + X	Dy/dx = 6X + 1	Dy/dx = 6X	Dy/dx = 3X + 1	Dy/dx = 6X^2 + x	
Find dy/dx of y = 6X^2 + 2X	Dy/dx = 12X + 2	Dy/dx = 12X	Dy/dx = 3X + 2	Dy/dx = 6X^2 + x	
Integrate with respect to X: y = 10X^3 - 4X + 1	(10/4)X^4 - 2X^2 + x				
Simplify 15ax^2/5x	3ax	3ax^2	5ax^2	5ax	
What is used to show differentiation of the equation y = f(x)?	dy/dx	dx/dy	y/x	dy/df	

## Test 14

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

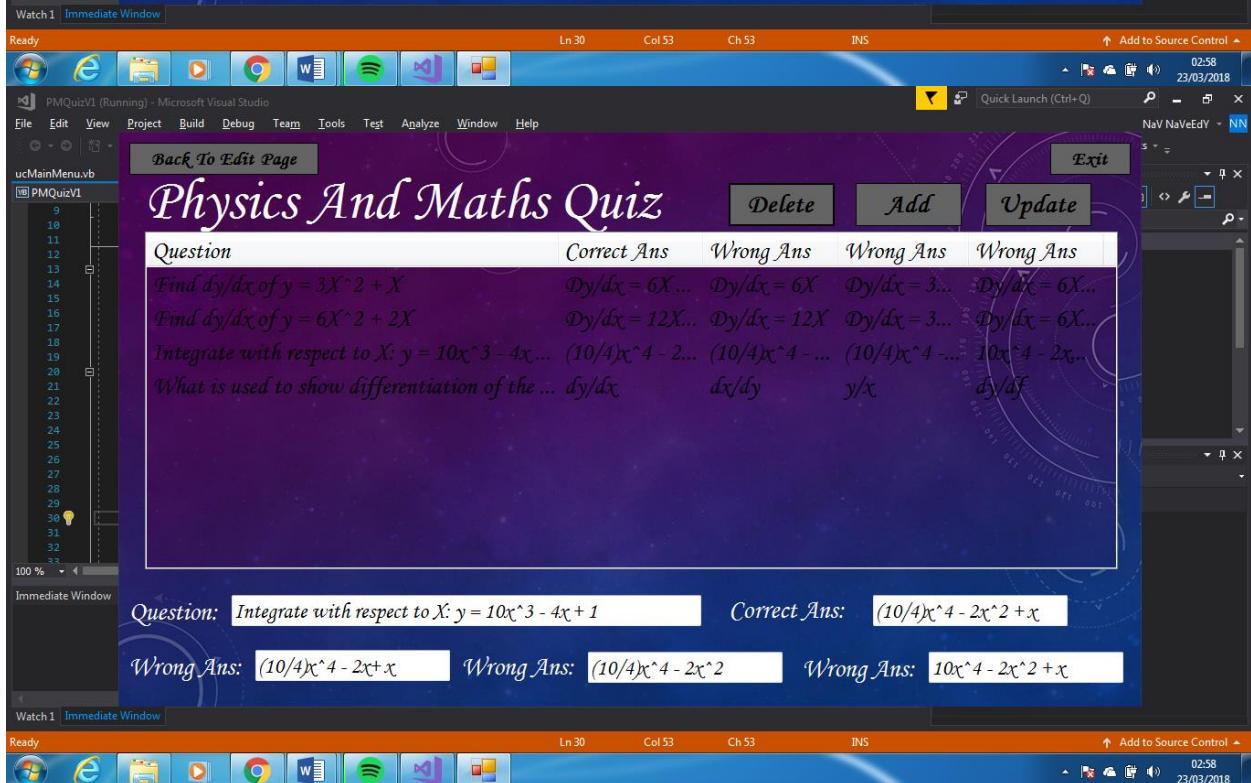
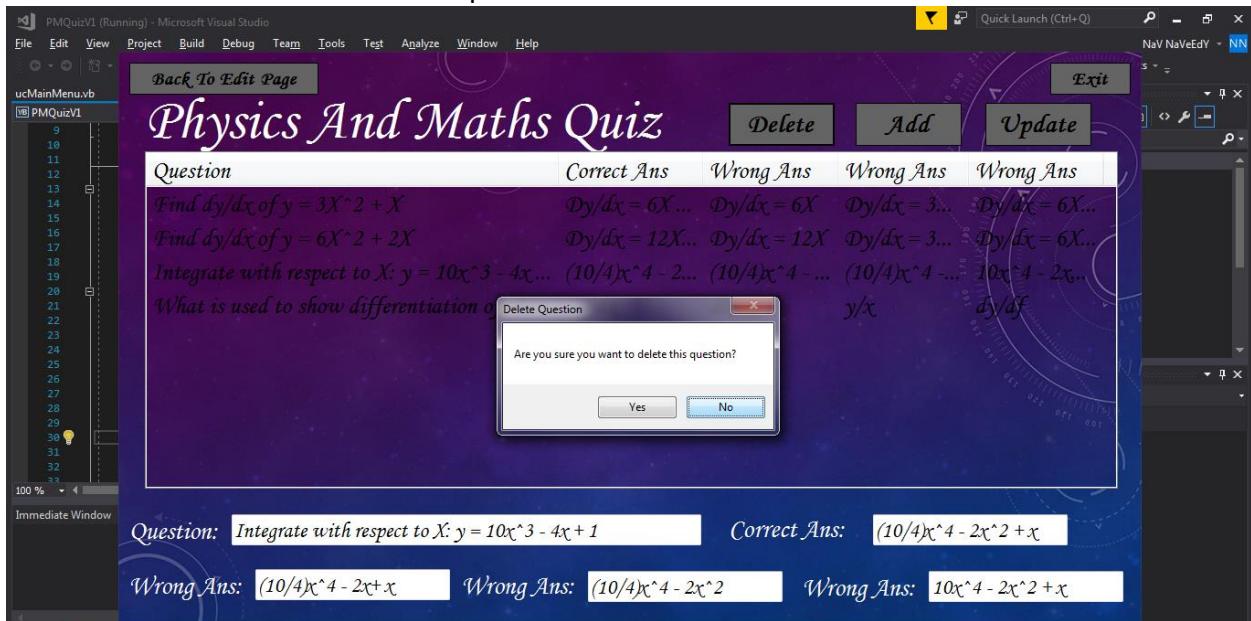


MQuestion	MA1	MA2	MA3	MA4	Click to Add
Find dy/dx of y = 3X^2 + X	Dy/dx = 6X + 1	Dy/dx = 6X	Dy/dx = 3X + 1	Dy/dx = 6X^2 + x	
Find dy/dx of y = 6X^2 + 2X	Dy/dx = 12X + 2	Dy/dx = 12X	Dy/dx = 3X + 2	Dy/dx = 6X^2 + x	
Integrate with respect to X: y = 10x^3 - 4x + 1	(10/4)x^4 - 2x^2 + x	(10/4)x^4 - 2x + x	(10/4)x^4 - 2x^2	10x^4 - 2x^2 + x	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	
What is used to show differentiation of the equation y = f(x)?	dy/dx	dx/dy	y/x	dy/df	
*					

## Test 15

Candidate Name: Naveed Ali Rafeeq

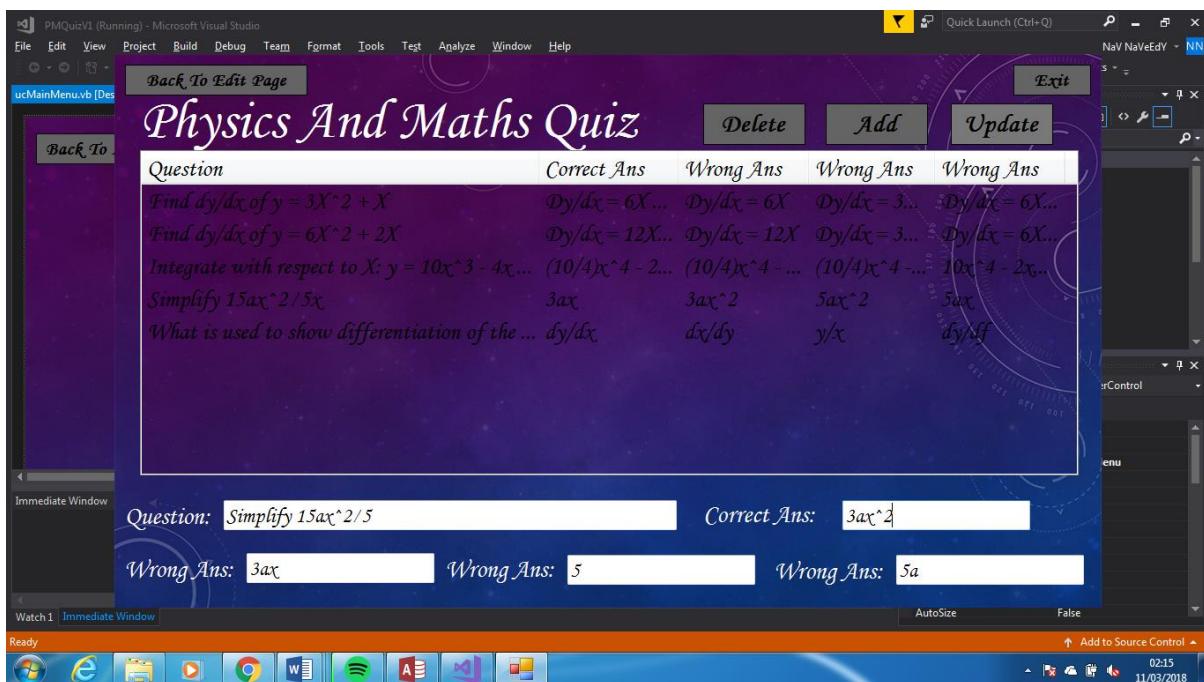
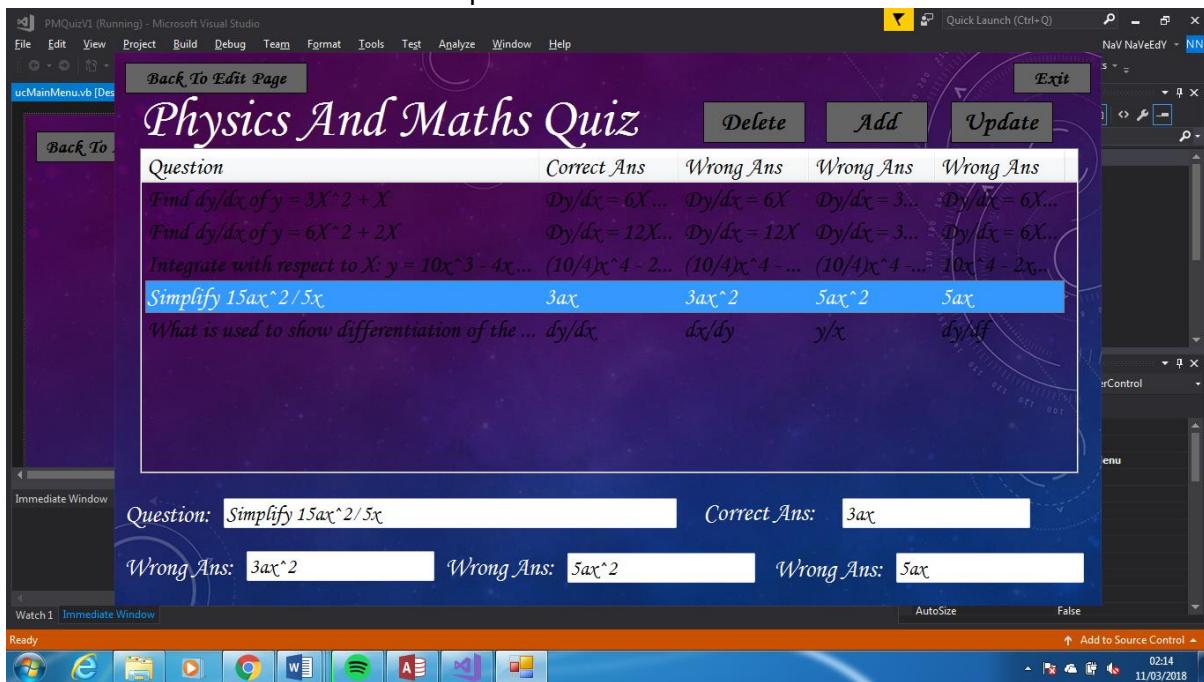
Candidate Number: 1904



## Test 16

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

**Physics And Maths Quiz**

Question	Correct Ans	Wrong Ans	Wrong Ans	Wrong Ans
Find $\frac{dy}{dx}$ of $y = 3x^2 + x$	$Dy/dx = 6x + 1$	$Dy/dx = 6x$	$Dy/dx = 3x$	$Dy/dx = 6x^2$
Find $\frac{dy}{dx}$ of $y = 6x^2 + 2x$	$Dy/dx = 12x + 2$	$Dy/dx = 12x$	$Dy/dx = 3x$	$Dy/dx = 6x$
Integrate with respect to X: $y = 10x^3 - 4x + 1$	$(10/4)x^4 - 2x^2 + C$	$(10/4)x^4 - 2x + 1$	$(10/4)x^4 - 2x^2$	$10x^4 - 2x_0 + 1$
Simplify $15ax^2/5$	$3ax^2$	$3ax$	$5$	$5a$
What is used to show differentiation of the equation $y = f(x)$ ?	$dy/dx$	$dx/dy$	$y/x$	$dy/df$

Question: Simplify  $15ax^2/5$

Correct Ans:  $3ax^2$

Wrong Ans:  $3ax$    Wrong Ans:  $5$    Wrong Ans:  $5a$

MQuestion	MA1	MA2	MA3	MA4	Click to Add
Find $\frac{dy}{dx}$ of $y = 3x^2 + x$	$Dy/dx = 6x + 1$	$Dy/dx = 6x$	$Dy/dx = 3x + 1$	$Dy/dx = 6x^2 + x$	
Find $\frac{dy}{dx}$ of $y = 6x^2 + 2x$	$Dy/dx = 12x + 2$	$Dy/dx = 12x$	$Dy/dx = 3x + 2$	$Dy/dx = 6x^2 + x$	
Integrate with respect to X: $y = 10x^3 - 4x + 1$	$(10/4)x^4 - 2x^2 + C$	$(10/4)x^4 - 2x + 1$	$(10/4)x^4 - 2x^2$	$10x^4 - 2x^2 + x$	
Simplify $15ax^2/5$	$3ax^2$	$3ax$	$5$	$5a$	
What is used to show differentiation of the equation $y = f(x)$ ?	$dy/dx$	$dx/dy$	$y/x$	$dy/df$	

## Test 17

**Physics And Maths Quiz**

Question	Correct Ans	Wrong Ans	Wrong Ans	Wrong Ans
Find $\frac{dy}{dx}$ of $y = 3x^2 + x$	$Dy/dx = 6x + 1$	$Dy/dx = 6x$	$Dy/dx = 3x$	$Dy/dx = 6x^2$
Find $\frac{dy}{dx}$ of $y = 6x^2 + 2x$	$Dy/dx = 12x + 2$	$Dy/dx = 12x$	$Dy/dx = 3x$	$Dy/dx = 6x$
Integrate with respect to X: $y = 10x^3 - 4x + 1$	$(10/4)x^4 - 2x^2 + C$	$(10/4)x^4 - 2x + 1$	$(10/4)x^4 - 2x^2$	$10x^4 - 2x^2 + x$
What is used to show differentiation of the equation $y = f(x)$ ?	$dy/dx$	$dx/dy$	$y/x$	$dy/df$

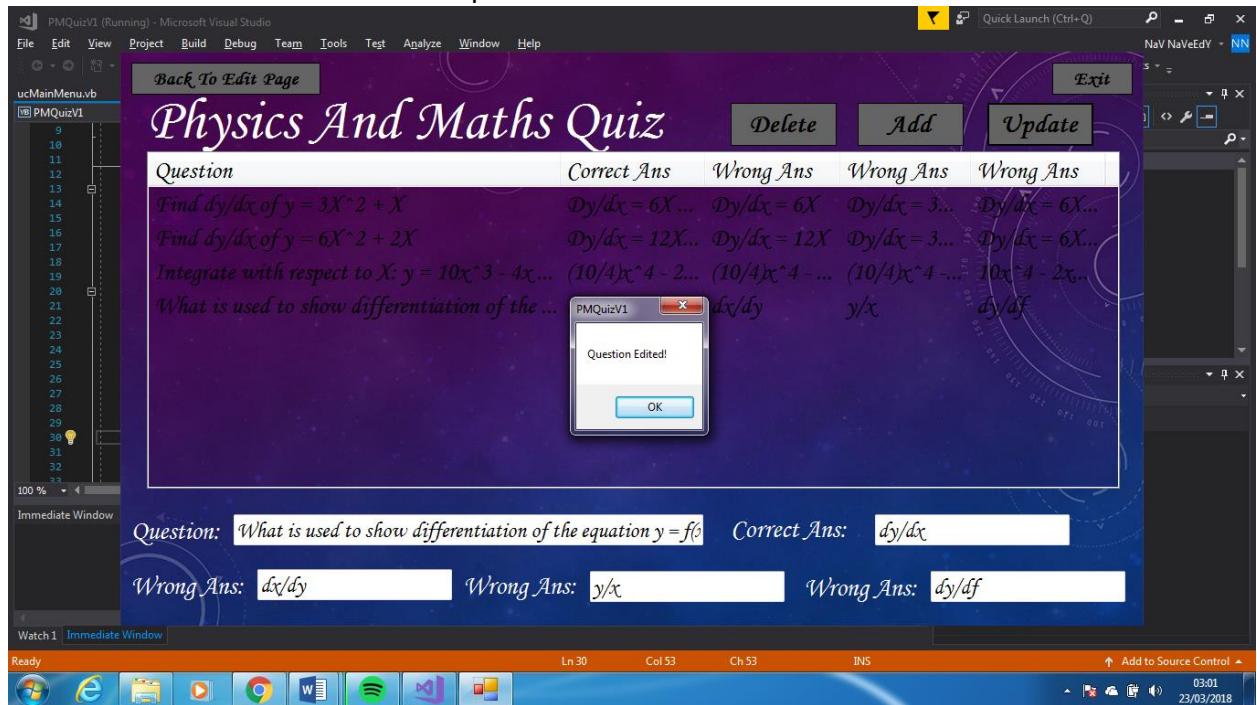
Question: What is used to show differentiation of the equation  $y = f(x)$ ?

Correct Ans:  $dy/dx$

Wrong Ans:  $dx/dy$    Wrong Ans:  $y/x$    Wrong Ans:  $dy/df$

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



## D. EVALUATION

### Testing

#### Logging In & Creating Users:

Test Number	What is being Tested?	Input/Action	Expected Outcome	Actual Outcome
1	Saves the users username and password to database.	<b>Valid:</b> Create new account button is pressed and a username and password are created.	When the user logs in with this username and password it should work.	<b>Valid:</b> The user can now login as many times as they want.
2	Allows the User to continue onto the main menu form if their login matches one in the database.	<b>Valid:</b> Input the login details of an already created student or teacher account.  <b>Invalid:</b> Input incorrect details into the username and password textboxes and click login.	<b>Valid:</b> When the user presses the enter button on password or clicks the login button it should direct them to the Main Menu.  <b>Invalid:</b> A message box should appear telling the user that their details are incorrect.	<b>Valid:</b> It directs them onto the Main Menu User Control  <b>Invalid:</b> Same
3	When creating a new user, it checks whether the password and the confirm password textboxes are	<b>Valid:</b> Both the textboxes have the same case-sensitive characters.	<b>Valid:</b> The program carries on the create user sub as usual.	<b>Valid:</b> Sub continues

	the same.	<b>Invalid extreme:</b> Both the text boxes have the same characters but not all of them are in the same case.	<b>Invalid:</b> A message box informing the users that the passwords are wrong and then exit the sub.	<b>Invalid:</b> The sub carries on it only stops if the passwords are different.
<b>4</b>	When creating a new user, it checks whether the username within the username textbox has already been taken.	<b>Valid:</b> The characters within the textbox is not in the Students or Teachers tables.	<b>Valid:</b> The program carries on the create user sub as usual.	<b>Valid:</b> The program carries on the create user sub as usual.
		<b>Invalid:</b> The characters are already in one of the 2 tables.	<b>Invalid:</b> A message box informs the users that the username has already been taken and then exits the sub.	<b>Invalid:</b> A message box informs the users that the username has already been taken and then exits the sub.
<b>5</b>	When creating a new user, it checks whether the Teacher ID they inputted within the textbox matches one within the database.	<b>Valid:</b> The characters within the TID text box matches with one within the Teachers table.	<b>Valid:</b> The program creates the account within the student table and therefore they can now login.	<b>Valid:</b> The program creates the account within the student table and therefore they can now login.
		<b>Invalid:</b> The characters within the TID text box does not match with one within the Teachers	<b>Invalid:</b> A message box informs the users that the teacher ID is not valid and then exits the sub.	<b>Invalid:</b> A message box informs the users that the teacher ID is not valid and then exits the sub.

		table.		
6	How to log in into the admin main menu.	<p><b>Valid:</b> Click on the log in button with the correct teacher details.</p> <p><b>Invalid:</b> Click on the log in button with correct student details.</p>	<p><b>Valid:</b> The admin main menu form should load.</p> <p><b>Invalid:</b> The main menu form should load.</p>	<p><b>Valid:</b> The admin main menu User Control loads.</p> <p><b>Invalid:</b> The main menu User Control loads.</p>

### Navigation Between Forms:

Test Number	What is being Tested?	Input/Action	Expected Outcome	Actual Outcome
7	<p>When logging in, the user can jump into the main menu without having to press the log in button.</p> <p><b>Cannot be proved by a screenshot</b></p>	<p><b>Valid:</b> The user presses the enter button in the password textbox.</p> <p><b>Invalid Extreme:</b> The user presses the enter button in the password textbox but with wrong details.</p>	<p><b>Valid:</b> The main menu screen logs in if the user's login details are correct.</p> <p><b>Invalid extreme:</b> A message box appears informing the user that the details are incorrect.</p>	<p><b>Valid:</b> The main menu User Control loads if the user's login details are correct.</p> <p><b>Invalid extreme:</b> A message box appears informing the user that the details are incorrect.</p>
8	How to exit the program.	<p><b>Valid:</b> The user clicks on the "Exit" button in the top right of every form.</p> <p><b>Invalid</b></p>	<p><b>Valid:</b> A message box appears asking if they are sure they want to leave.</p>	<p><b>Valid:</b> A message box appears asking if they are sure they want to leave.</p>

		<b>Erroneous:</b> The user clicks on the form background.	<b>Invalid:</b> Nothing happens.	<b>Invalid:</b> Nothing happens.
9	How to go back to the login form from the main menu / create account form.	<p><b>Valid:</b> Click the "Login" button in the top left of the main menu and the create account forms</p> <p><b>Invalid:</b> Click the Exit button in the top right of the main menu / create account forms.</p>	<p><b>Valid:</b> The main menu / create account form closes and the login form loads.</p> <p><b>Invalid:</b> A message box loads asking if they are sure they want to leave.</p>	<p><b>Valid:</b> The main menu / create account UC closes and the login UC loads.</p> <p><b>Invalid:</b> A message box loads asking if they are sure they want to leave.</p>
10	How to exit a quiz but stay on the program.	<p><b>Valid:</b> Click on the back button in the top left of a quiz, and then click yes on the message box that appears.</p> <p><b>Invalid Extreme:</b> Click on the back button in the top left of the quiz, and then click NO on the message box that appears.</p>	<p><b>Valid:</b> The main menu form should load on the program.</p> <p><b>Invalid Extreme:</b> It should return you to your quiz and the same question that the user was currently on.</p>	<p><b>Valid:</b> The main menu UC loads.</p> <p><b>Invalid Extreme:</b> You return to your quiz and the same question that the user is currently on.</p>
11	How to start a maths / physics quiz.	<b>Valid:</b> Whilst on the main menu form click on the	<b>Valid:</b> The corresponding quiz should load on the	<b>Valid:</b> The corresponding quiz UC loads on the program

		maths / physics quiz button.	program with the sub title of Question: 1 / 20.	with the sub title of Question: 1 / 20.
		<b>Invalid:</b> Whilst on the main menu form click on the about button.	<b>Invalid:</b> The about form should load with the details of the quiz.	<b>Invalid:</b> The about UC loads with the details of the quiz.
12	How to view the leader boards as a student.	<b>Valid:</b> Click on the view leader boards button on the feedback form  <b>Invalid:</b> Click on the back to main menu button on the feedback form.	<b>Valid:</b> A list view of the students ranked by scores should load.  <b>Invalid:</b> The main menu should load.	<b>Valid:</b> A list view of the students ranked by scores loads and the feedback becomes hidden.  <b>Invalid:</b> The main menu UC loads.
13	How to view the leader boards as a teacher.	<b>Valid:</b> Click on the leader boards button on the admin menu form.  <b>Invalid:</b> Click on the back to login button in the top left of the admin menu form.	<b>Valid:</b> A list view of the students ranked by scores should load.  <b>Invalid:</b> The teacher should be signed out and the login form should load.	<b>Valid:</b> List views of the students ranked by scores load, one Maths and the other Physics.  <b>Invalid:</b> The teacher is signed out and the login UC loads.
14	How to go to the next question in the	<b>Valid:</b> Click on the next button after clicking the	<b>Valid:</b> The next question will load on the same form and	<b>Valid:</b> There is no next button The UC refreshes on the same

Candidate Name: Naveed Ali Rafeeq Candidate Number: 1904

	quizzes.	answer button.	the question number will become incremented.	form and the question number becomes incremented, when the answer button is selected
		<b>Invalid Extreme:</b> Click on the next button before clicking on the answer button.	<b>Invalid Extreme:</b> Nothing will happen due to the answer button not being clicked first.	<b>Invalid Extreme:</b> The UC refreshes on the same form and the question number becomes incremented.

### Maths / Physics Quiz:

Test Number	What is being Tested?	Input/Action	Expected Outcome	Actual Outcome
15	Answering a question after clicking on one of the four radio buttons.	<b>Valid:</b> Click on the answer button.  <b>Invalid Erroneous:</b> Press the enter key	<b>Valid:</b> A message box will load.  <b>Invalid Erroneous:</b> Nothing happens.	<b>Valid:</b> A message box loads with a correct or incorrect message.  <b>Invalid Erroneous:</b> Nothing happens
16	Answering a question without clicking on one of the four radio buttons first.	<b>Valid:</b> Click on the answer button.  <b>Invalid Erroneous:</b> Press the right arrow key.	<b>Valid:</b> Nothing happens.  <b>Invalid Erroneous:</b> Nothing happens	<b>Valid:</b> The UC refreshes on the same form and the question number becomes incremented.  <b>Invalid Erroneous:</b> Nothing happens
17	Choosing a correct answer.	<b>Valid:</b> Click the right radio button corresponding to the question and then click the	<b>Valid:</b> A message box should load with the text correct, and then the right answer and question	<b>Valid:</b> A message box loads with the text correct, and then the right answer and question counter are

		answer button.	counter should both increment.	both incremented.
		<b>Invalid:</b> Click the right radio button corresponding to the question and then click the next question button.	<b>Invalid:</b> Nothing happens.	<b>Invalid Erroneous:</b> There is no next question button
18	Choosing an incorrect answer	<b>Valid:</b> Click the wrong radio button corresponding to the question and then click the answer button.  <b>Invalid</b> <b>Erroneous:</b> Press the enter button after choosing a radio button.	<b>Valid:</b> A message box should load with the text incorrect, and then the question counter should increment.  <b>Invalid:</b> Nothing happens.	<b>Valid:</b> A message box loads with the text incorrect, and then the question counter is incremented.  <b>Invalid:</b> Nothing happens.
19	The correct title loads with its corresponding quiz Maths / Physics.	<b>Valid:</b> Click on a quiz in the main menu.  <b>Invalid Extreme:</b> Click on the about button.	<b>Valid:</b> Maths / Physics quiz appears at the top of the program.  <b>Invalid:</b> About appears on the top of the program.	<b>Valid:</b> Maths / Physics quiz appears at the top of the program.  <b>Invalid:</b> About appears on the top of the program.
20	How to complete the quiz.	<b>Valid:</b> Answer 20 questions.  <b>Invalid Extreme:</b> Answer 19 questions.	<b>Valid:</b> The feedback form of the program will load.  <b>Invalid:</b> The last question of the quiz	<b>Valid:</b> The corresponding feedback UC will load.  <b>Invalid:</b> The last question of the quiz is

			will be displayed.	displayed.
21	Question structure	Complete the quiz	No question should be repeated.	Sometimes the question is repeated due to there not being enough questions in the database.
22	Answer structure	Complete the quiz	All the answers to a question should be different to each other and should always appear in a random order.	All the answers are different to each other and always appears in a random order
23	The question number should be reset when loading the quiz.	<b>Valid:</b> Every time the main menu is loaded.  <b>Invalid:</b> Click on main menu button in the top left and then click no on the message box.	<b>Valid:</b> The question counter returns to 0.  <b>Invalid:</b> The question counter stays unchanged therefore so does the question number.	<b>Valid:</b> The question counter returns to 0.  <b>Invalid:</b> The question counter stays unchanged therefore so does the question number.
24	The right answer counter should reset when leaving the quiz.  <b>Cannot be proved by a screenshot</b>	<b>Valid:</b> Every time the main menu is loaded.  <b>Invalid:</b> The exit button is clicked and then the user clicks no on the message box.	<b>Valid:</b> The right answer counter returns to 0.  <b>Invalid:</b> The quiz will continue from where it was previously.	<b>Valid:</b> The right answer counter returns to 0.  <b>Invalid:</b> The quiz continues from where it was previously.
25	An even number of	Complete the quiz	Roughly half of the questions are from	This was not implemented in the

	times a question with random values appears.  <b>Was not implemented into the development</b>		the random number generated.	development.
--	-----------------------------------------------------------------------------------------------------	--	------------------------------	--------------

**Feedback & Leader Boards:**

Test Number	What is being Tested?	Input/Action	Expected Outcome	Actual Outcome
26	The leader boards are ranked by the scores of each student in a class.	<b>Valid:</b> Click on the leader boards button.  <b>Invalid:</b> Click on the main menu button in the top left of the feedback UC.	<b>Valid:</b> A sort algorithm should sort out the order of the scores in a list view  <b>Invalid:</b> The main menu form should load.	<b>Valid:</b> The SQL statement orders the all the scores in descending order all with the same TID.  <b>Invalid:</b> The main menu UC loads.
27	The values in the leader board cannot be changed.	<b>Valid:</b> Click on a user and type in characters on the keyboard.  <b>Invalid:</b> Click on the main menu button in the top left.	<b>Valid:</b> Nothing happens.  <b>Invalid:</b> The main menu form loads.	<b>Valid:</b> The username becomes highlighted.  <b>Invalid:</b> The main menu UC loads.
28	Feedback User	Complete the	The Feedback UC	The Feedback UC

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

	Control displays the users score.	quiz	should display the total amount of questions the user got right out of 20.	displays the total amount of questions the user got right out of 20 as well as a grade and a message to go with it.
--	-----------------------------------	------	----------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------

### Editing Questions & Accounts

Test Number	What is being Tested?	Input/Action	Expected Outcome	Actual Outcome
29	The teacher can add questions to the quiz.	<p><b>Valid:</b> The teacher types a question and four answers into separate textboxes and clicks add question.</p> <p><b>Invalid Extreme:</b> The teacher clicks the add question button without typing into the textboxes.</p>	<p><b>Valid:</b> The new question is added to the M/P Questions table.</p> <p><b>Invalid:</b> A message box appears informing the teacher that the textboxes are blank.</p>	<p><b>Valid:</b> The new question is added to the M/P Questions table.</p> <p><b>Invalid:</b> The new 'question' is added to the M/P Questions table.</p>
30	The teacher can edit questions within the P / M Question table.	<p><b>Valid:</b> The teacher clicks the update.</p> <p><b>Invalid:</b> The teacher clicks the update button with empty textboxes</p>	<p><b>Valid:</b> The question and answers should be changed to the ones in the textboxes.</p> <p><b>Invalid:</b> Nothing should happen.</p>	<p><b>Valid:</b> The question and answers are changed to the ones in the textboxes and a message box informs the user.</p> <p><b>Invalid:</b> Nothing happens.</p>

<b>31</b>	The teacher can delete an account from their class.	<b>Valid:</b> The teacher clicks on the User List button and then clicks on the user they wish to delete, then click the delete button.	<b>Valid:</b> The user is now deleted from the Students table in the database, they can no longer log in.	<b>Valid:</b> The user is now deleted from the Students table in the database, they can no longer log in.
		<b>Invalid Extreme:</b> The teacher clicks on the User List button and then clicks on the user they wish to delete.	<b>Invalid Extreme:</b> Nothing happens.	<b>Invalid Extreme:</b> Nothing happens.

I decided to gather my stake holders Moaz and Brendan and asked them to beta test my program as a whole, this is their feedback.

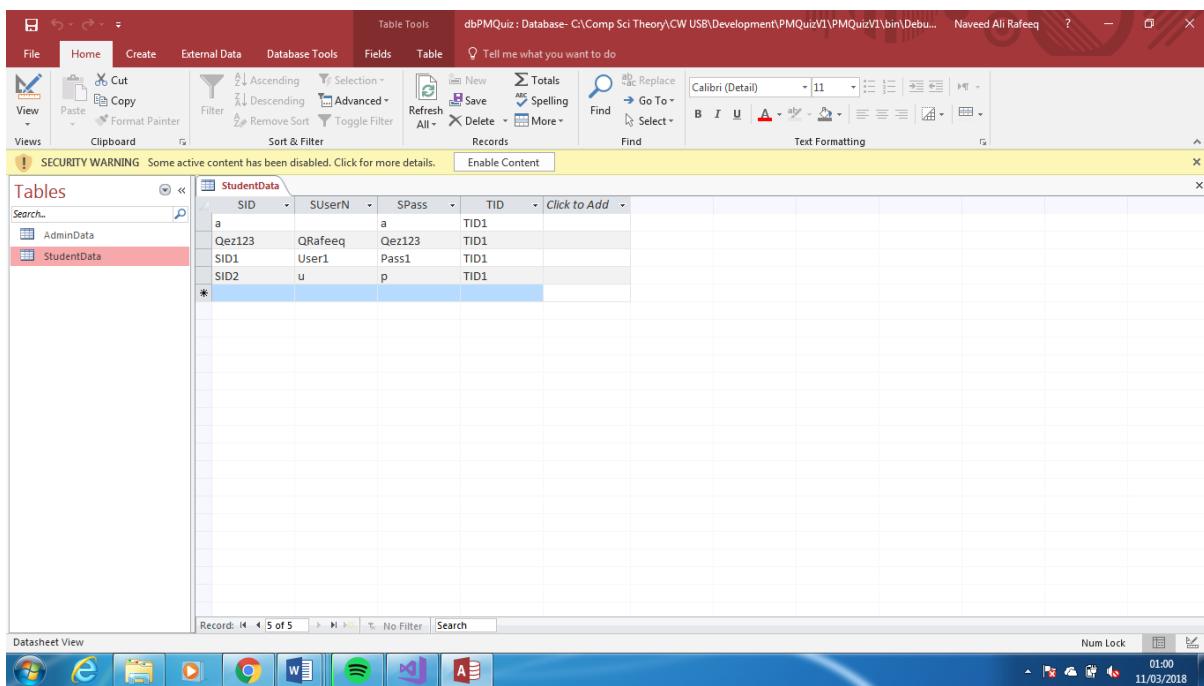
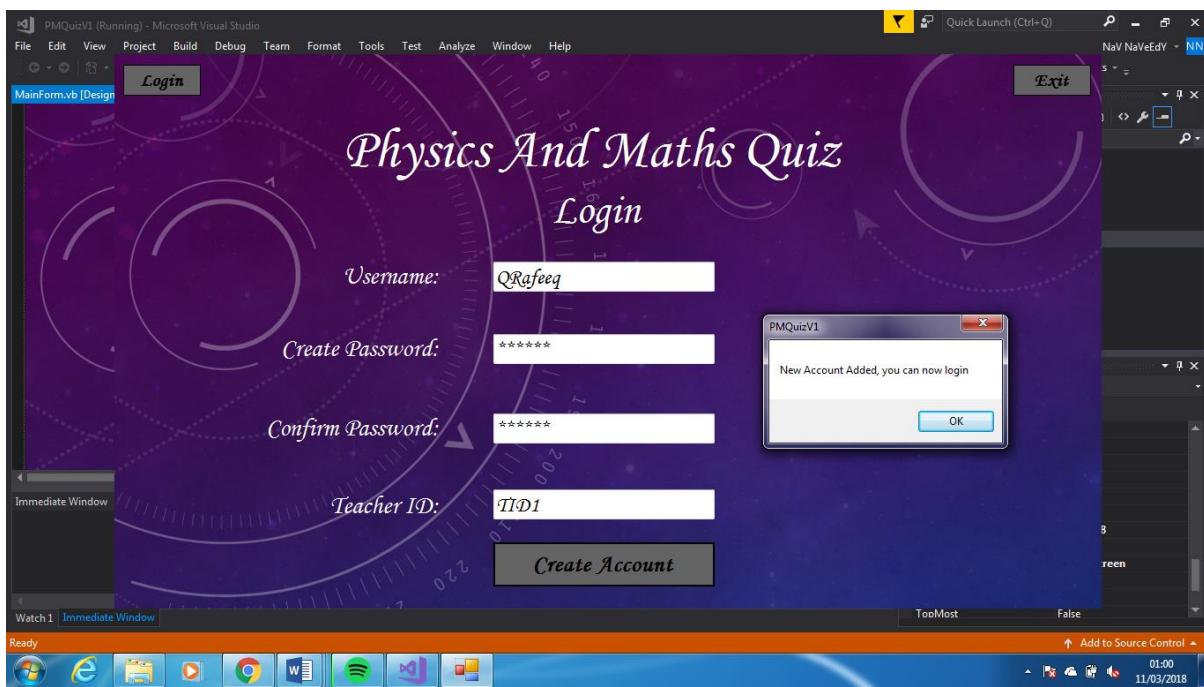
It was quite easy to manoeuvre around the program, the User Controls allowed easy progression to different parts of the Quiz since they manage to stack on each other or be easily disposed and loaded. The exit button was in the same place no matter which section you were on, which meant it was easily accessible since everyone knew that it was in the top right corner. The back buttons where also in the same place as well, even though they were different buttons the general ‘back’ button was usually always placed in the top left. One problem however was the fact that nothing stacked on each other, it was kind of like a power point the way you would just go from User Control to User Control. You could have created some panels that could stack onto your user control but not be the same size, this was you get a more in depth and interactive navigation throughout the Program. Nearly everything within the program worked as it needed it, there was some changes implemented that was not agreed upon when we were setting out the success criteria however, I believe that most were for the best. Like from switching from a form base program to one form base and multiple User Controls was definitely an improvement. All in all, I guess we are happy with what has been developed within the program since it is usable, but it would be best if you developed it further so that other features could be added. Like topics within each subject and how the amount of questions change depending on how well you do in each subject.

## **Testing Evidence**

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

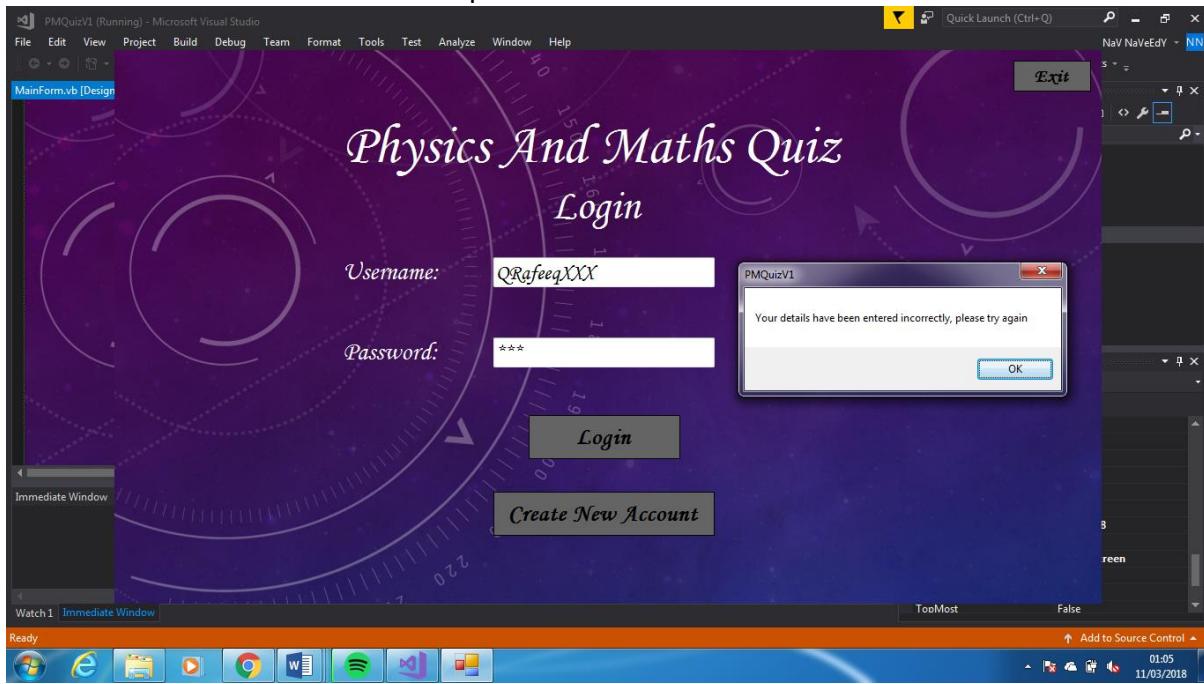
**Test 1:**



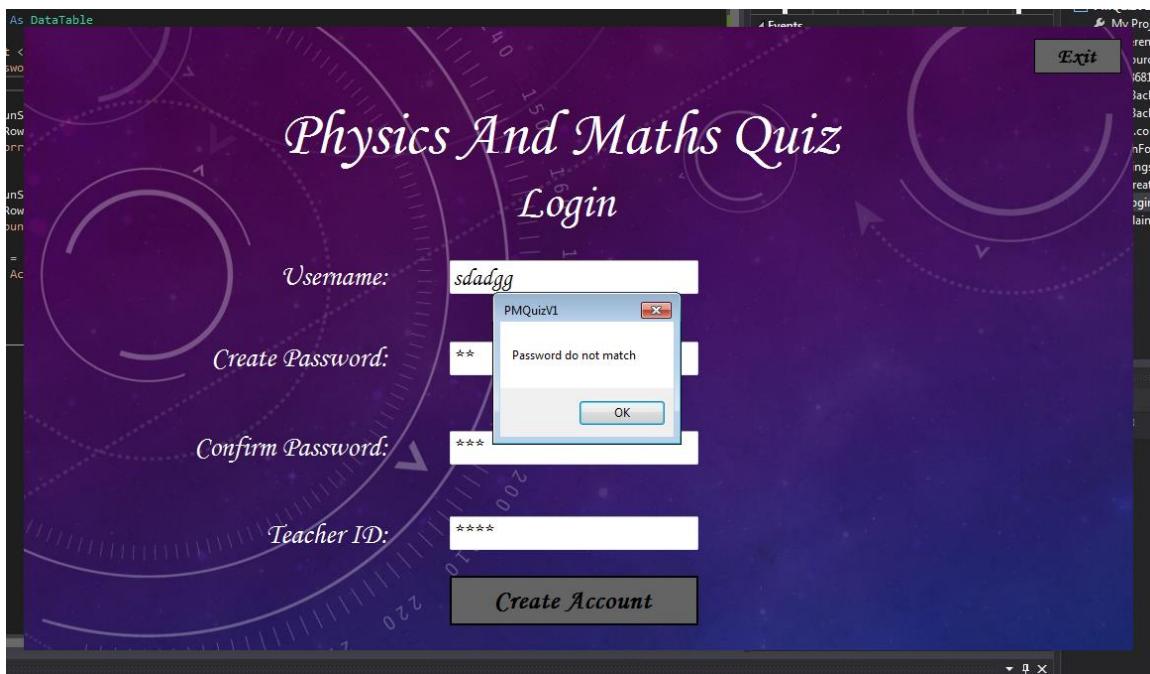
**Test 2:**

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



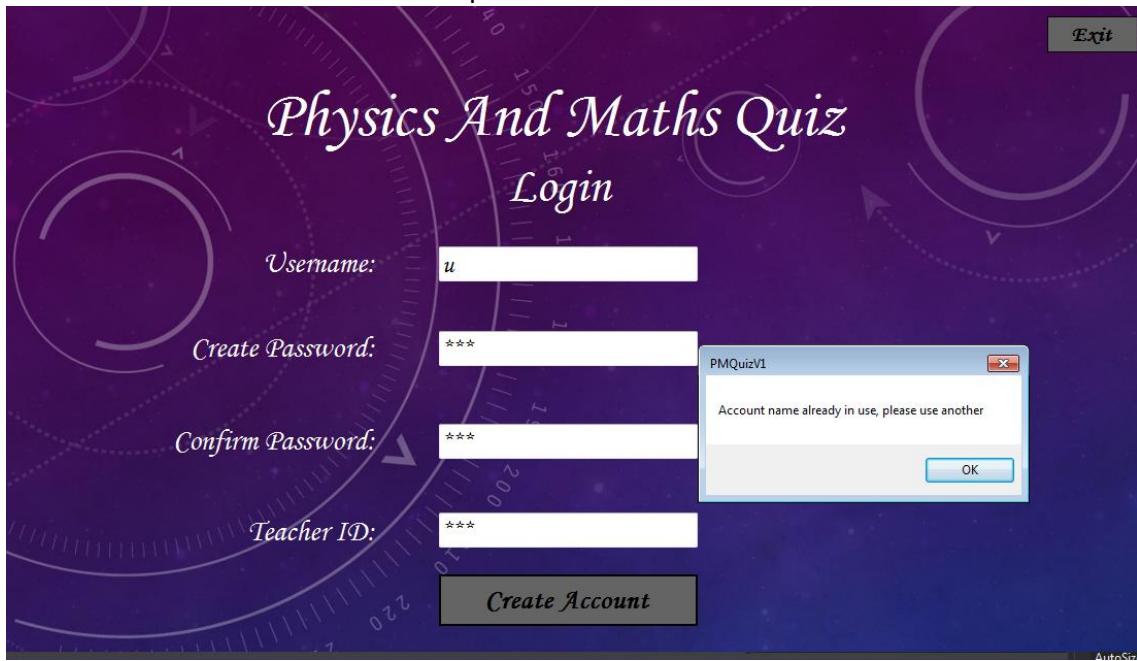
### Test 3:



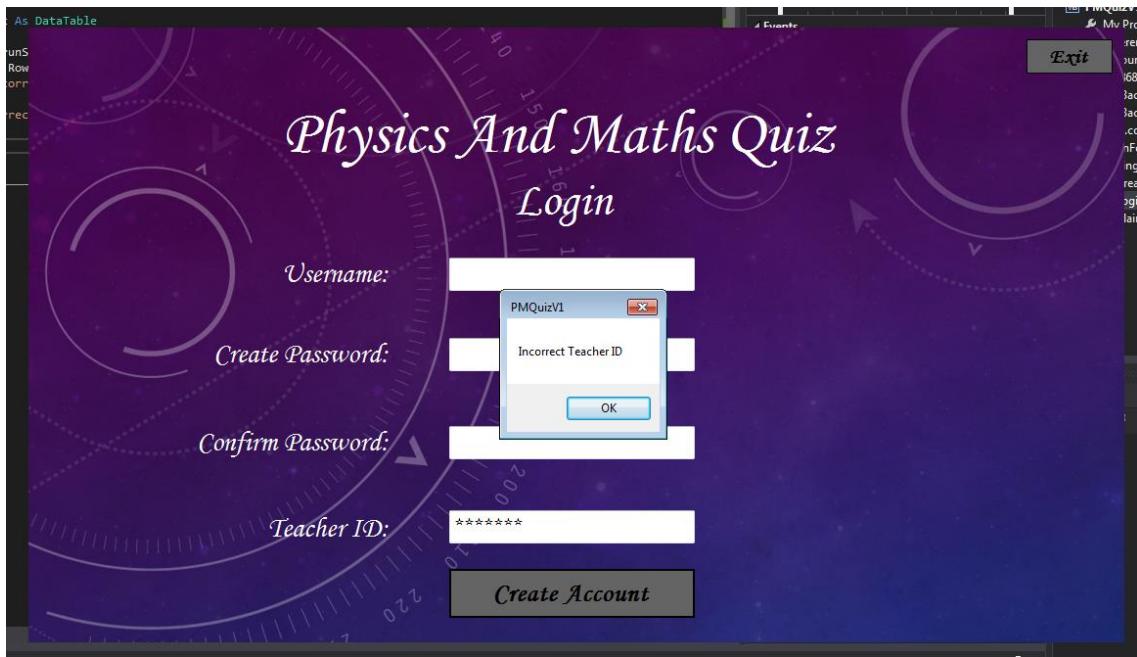
### Test 4:

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



**Test 5:**



**Test 6:**

Candidate Name: Naveed Ali Rafeeq

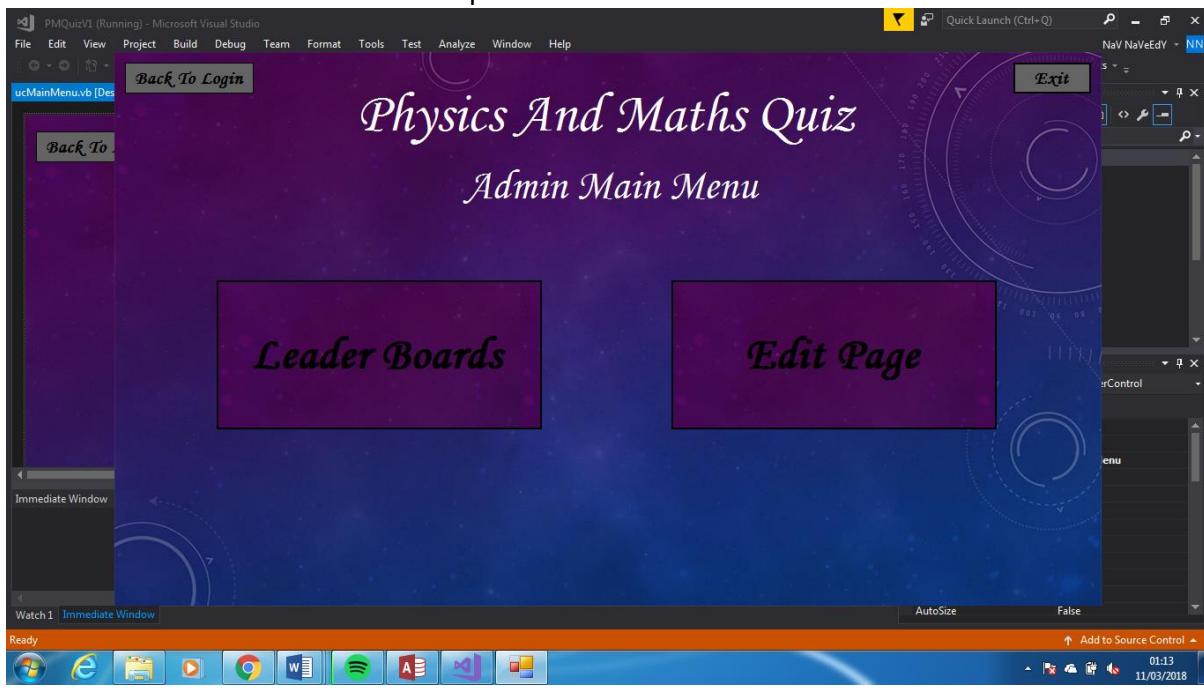
Candidate Number: 1904

The screenshot shows two windows side-by-side. The left window is a Microsoft Access application displaying the 'AdminData' table in Datasheet View. The table has columns: TID, TUserName, and TPass. It contains two records: TID1 (BrendanFoster, password) and TID2 (T1, p). The right window is a custom application titled 'Physics And Maths Quiz Login'. It features a purple background with a circular clock face design. It includes fields for 'Username:' (containing 't1') and 'Password:' (containing 'p'), a 'Login' button, and a 'Create New Account' button. The taskbar at the bottom shows various icons, and the system tray indicates the date and time as 11/03/2018 and 01:12.

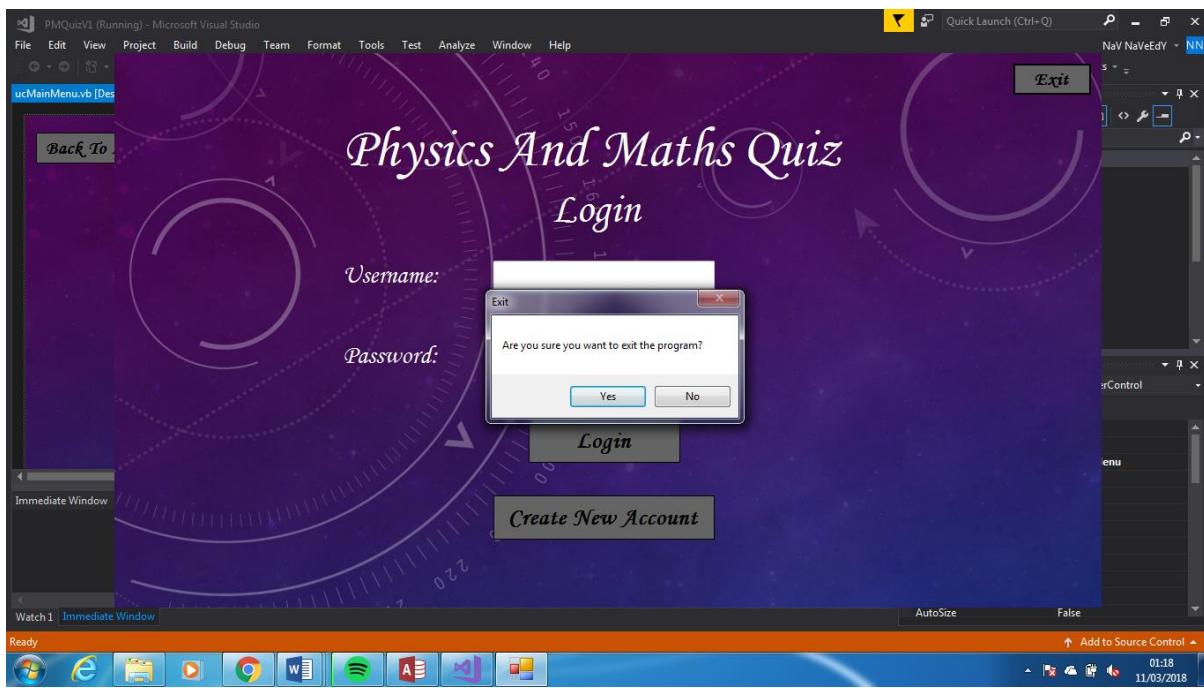
TID	TUserName	TPass
TID1	BrendanFoster	password
TID2	T1	p

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



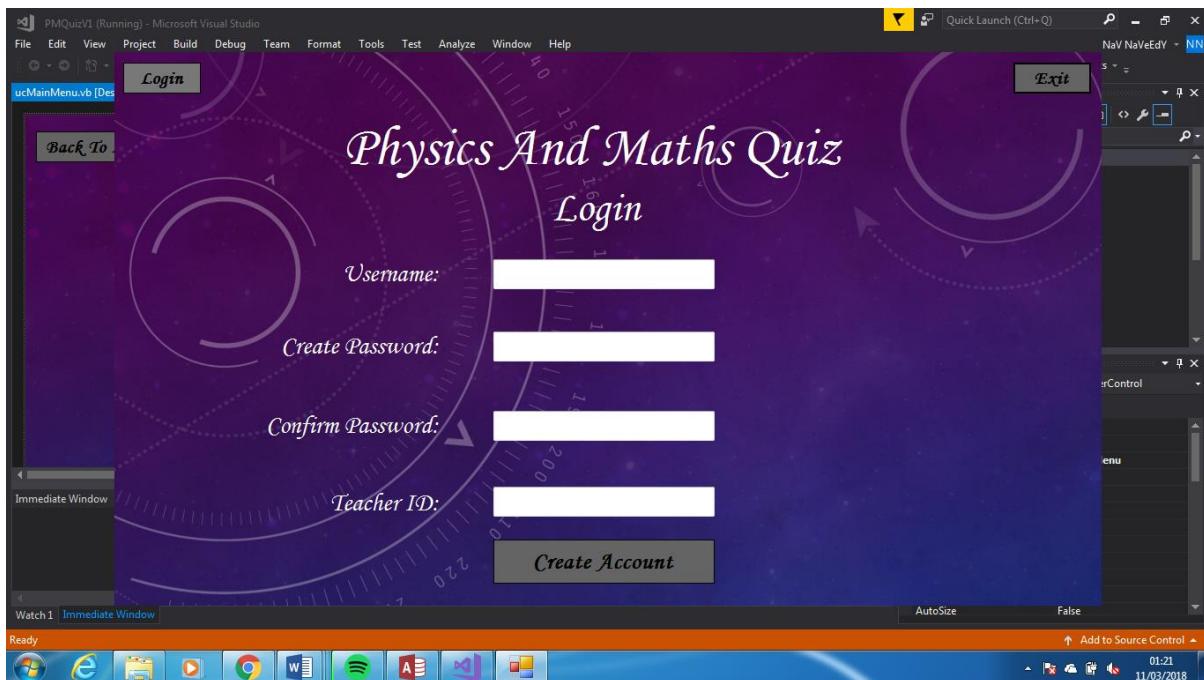
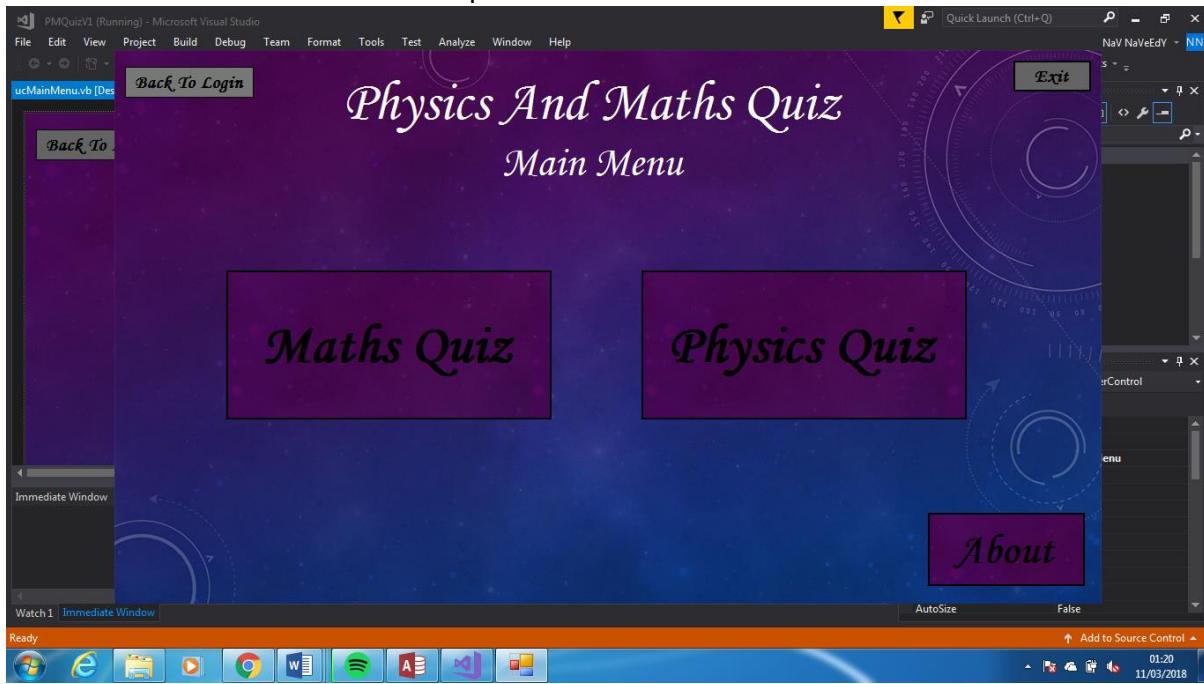
### Test 8:



### Test 9:

Candidate Name: Naveed Ali Rafeeq

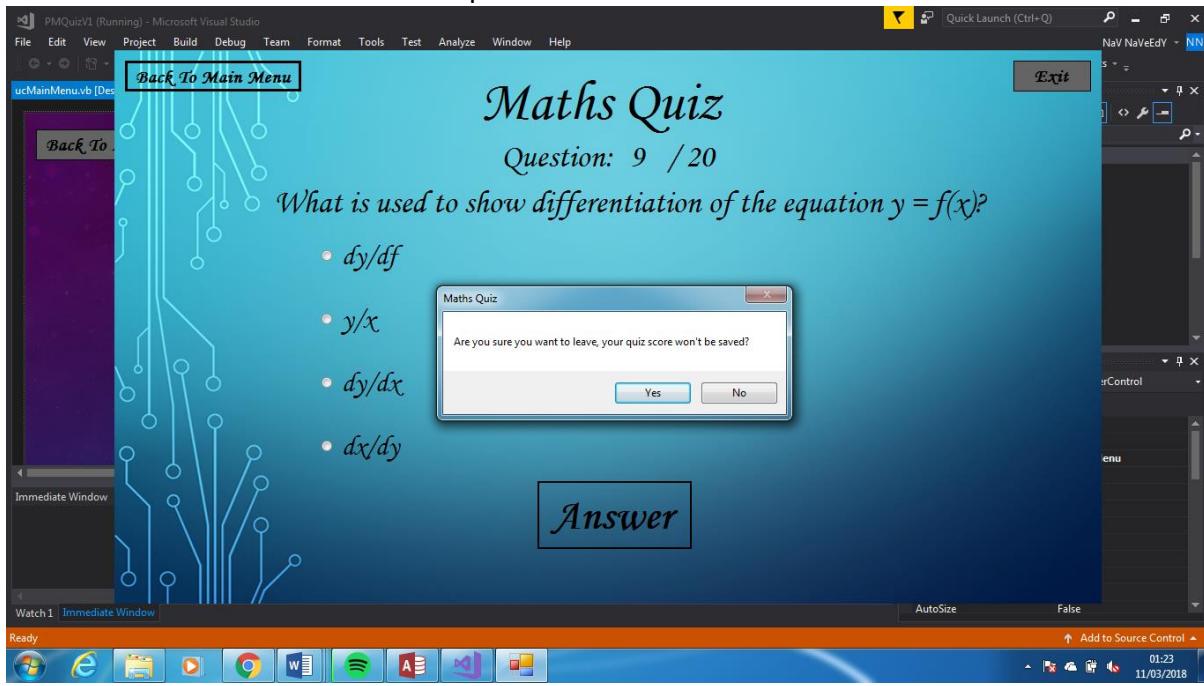
Candidate Number: 1904



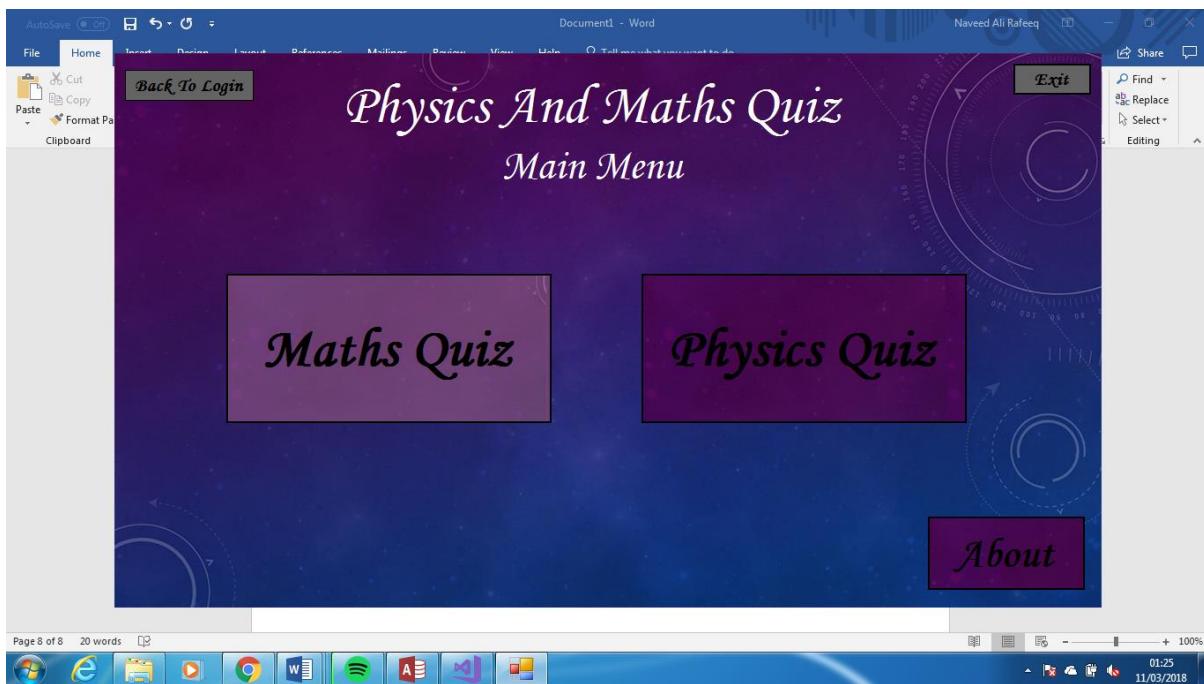
**Test 10:**

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

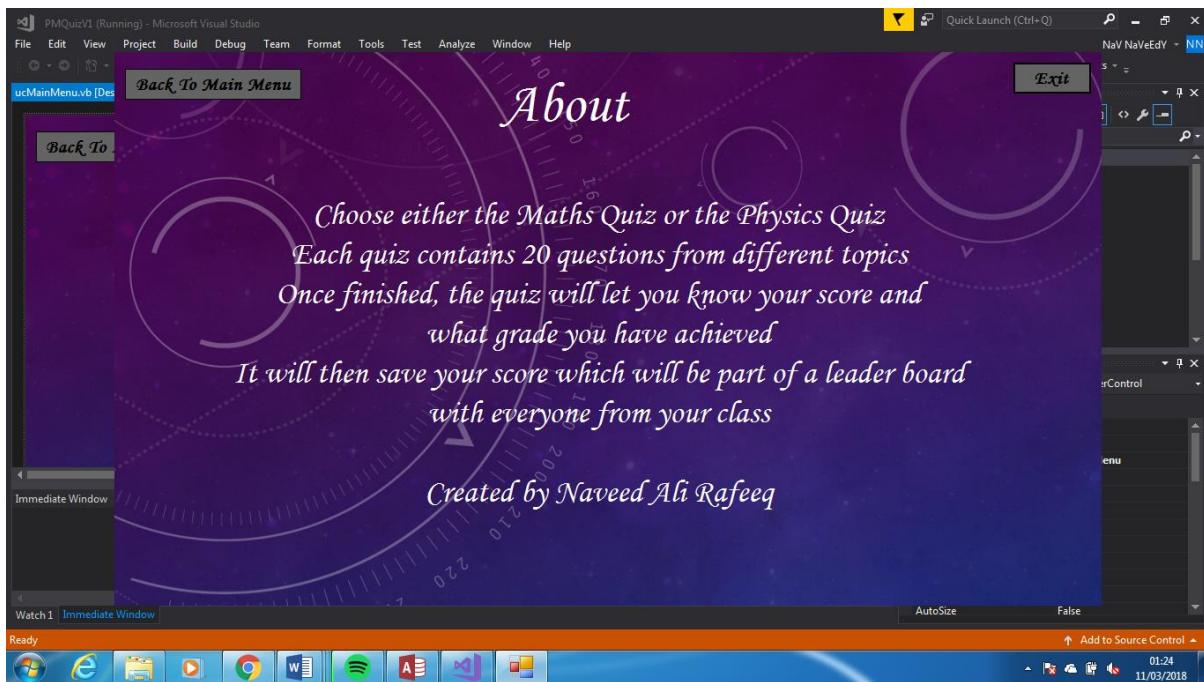
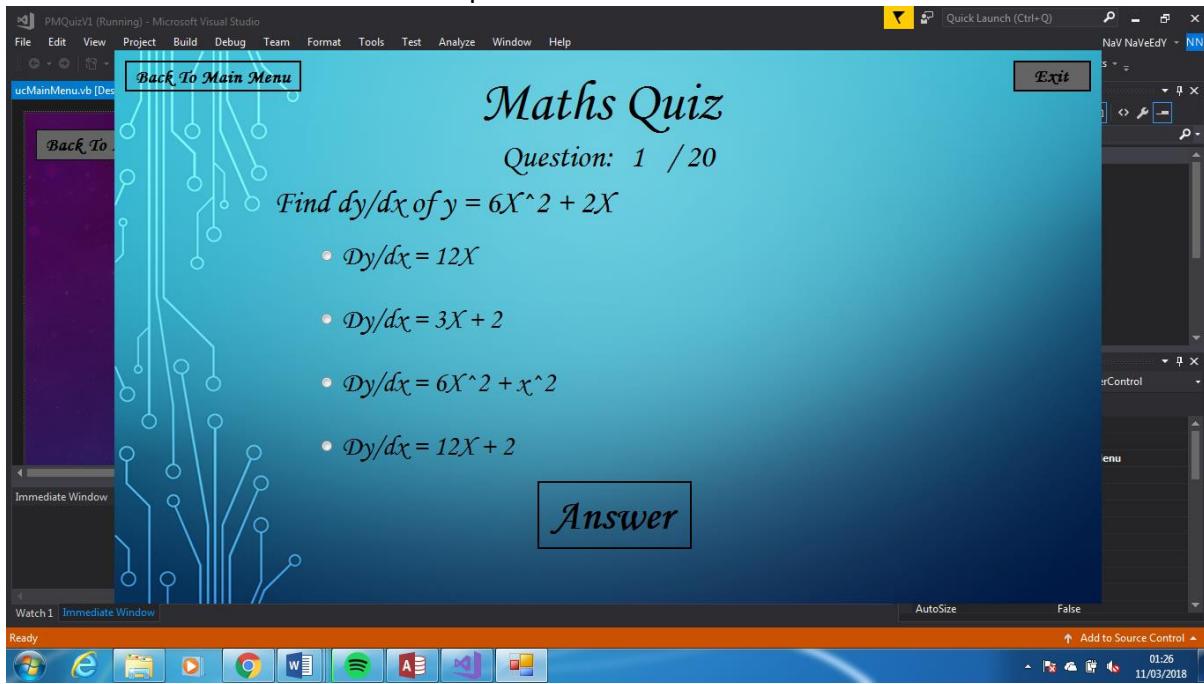


### Test 11:



Candidate Name: Naveed Ali Rafeeq

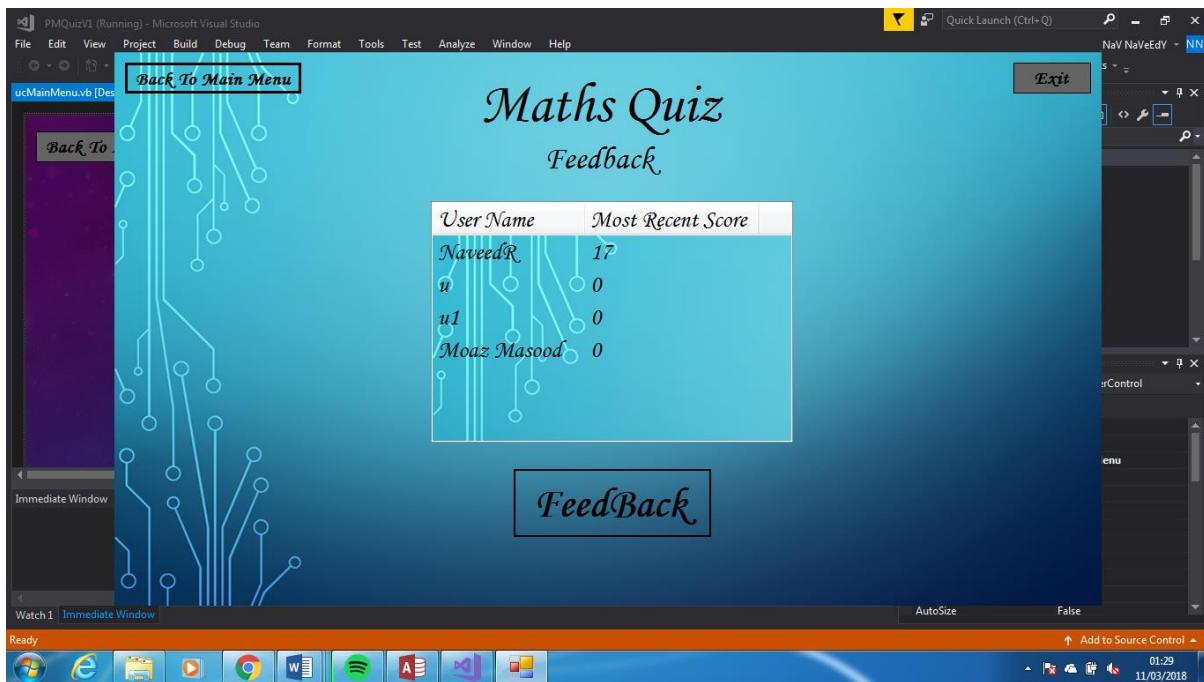
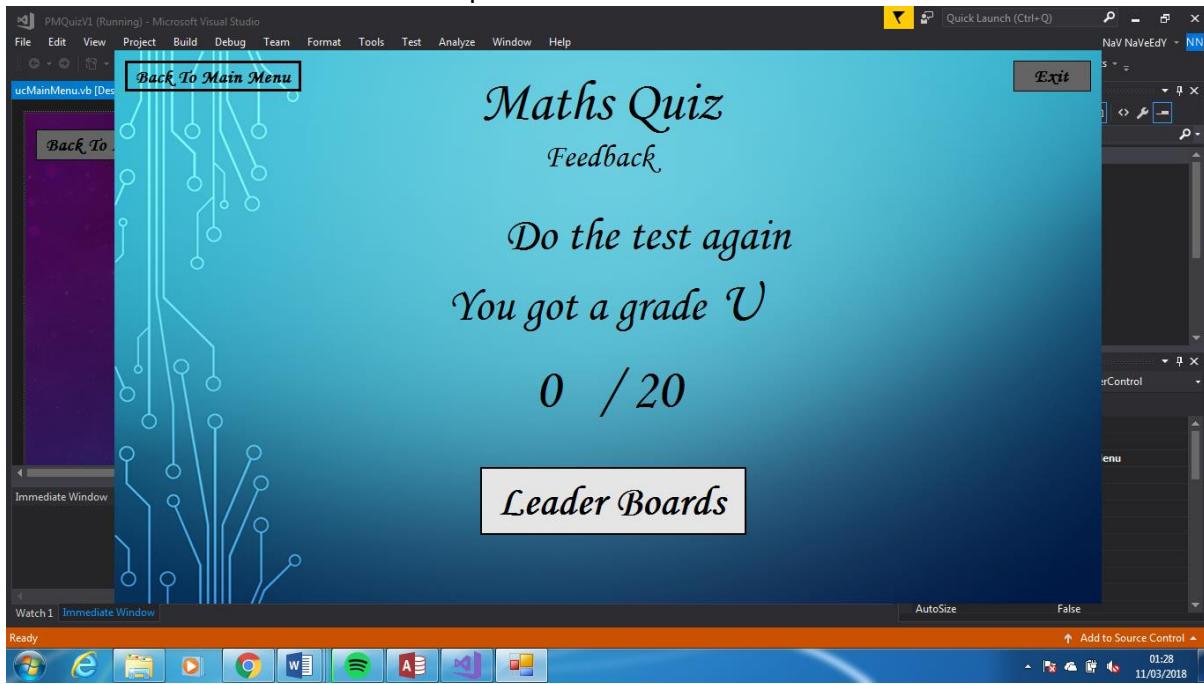
Candidate Number: 1904



**Test 12:**

Candidate Name: Naveed Ali Rafeeq

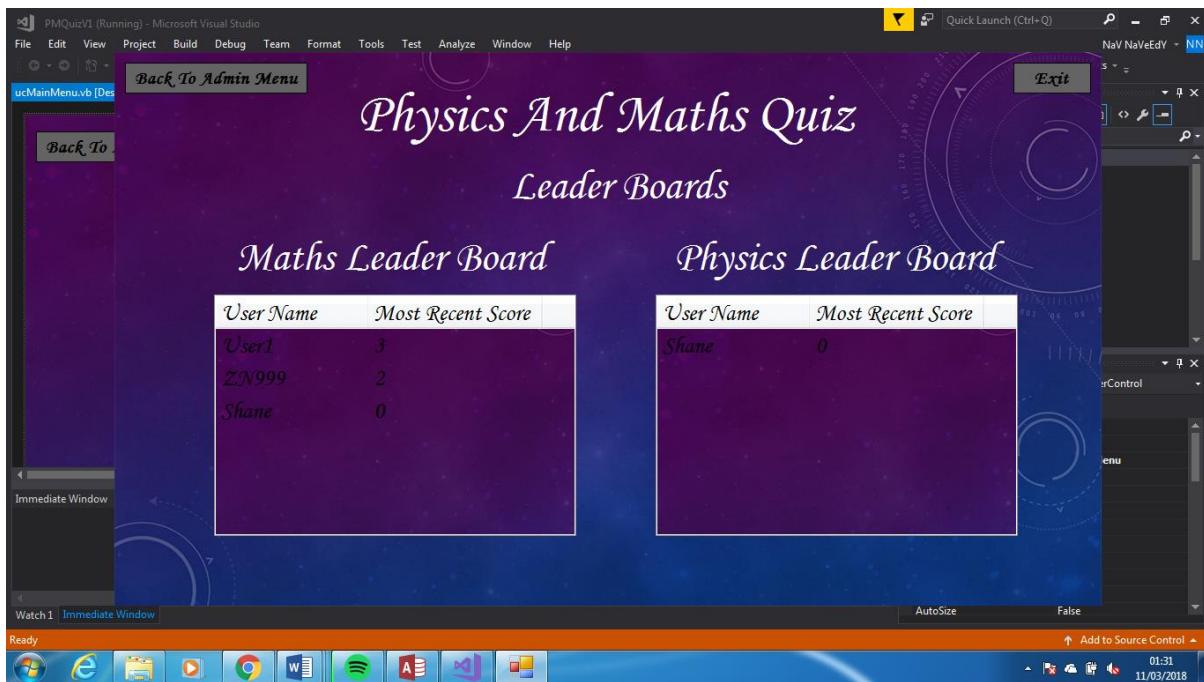
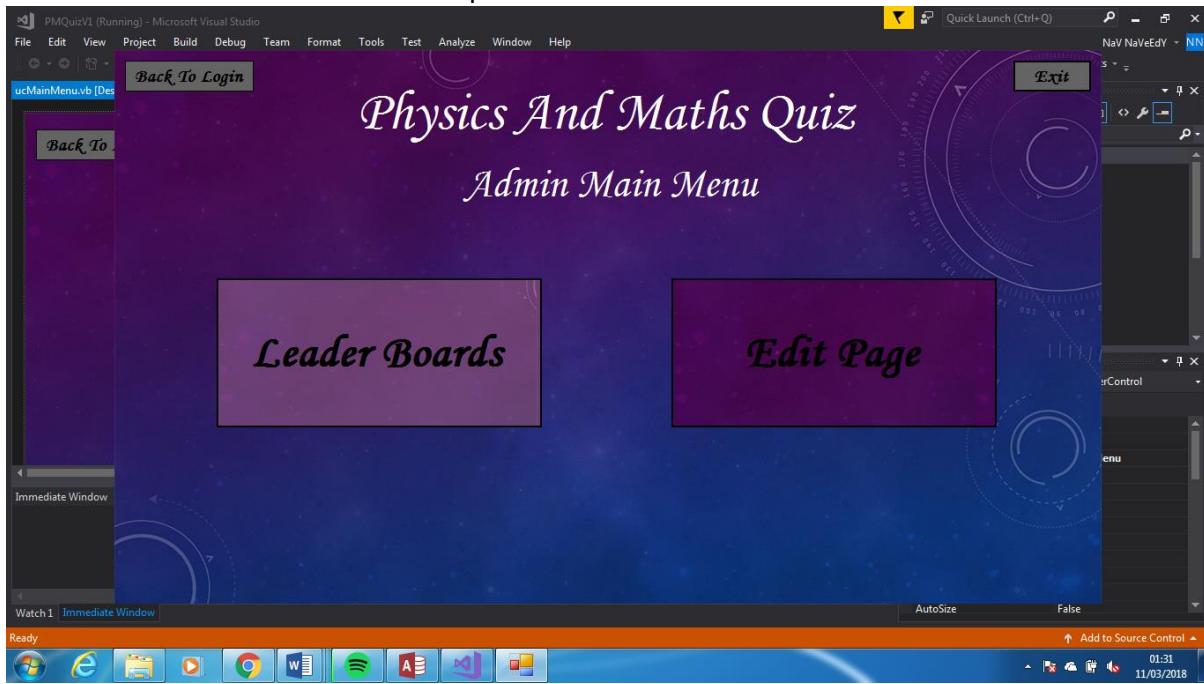
Candidate Number: 1904



**Test 13:**

Candidate Name: Naveed Ali Rafeeq

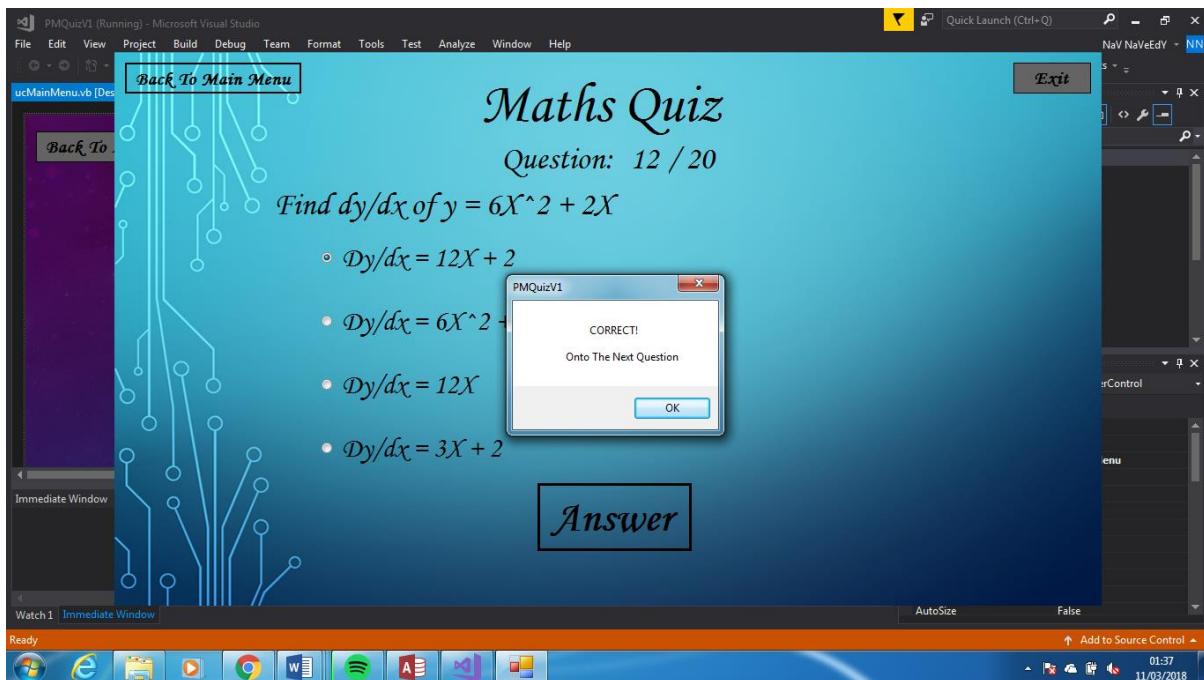
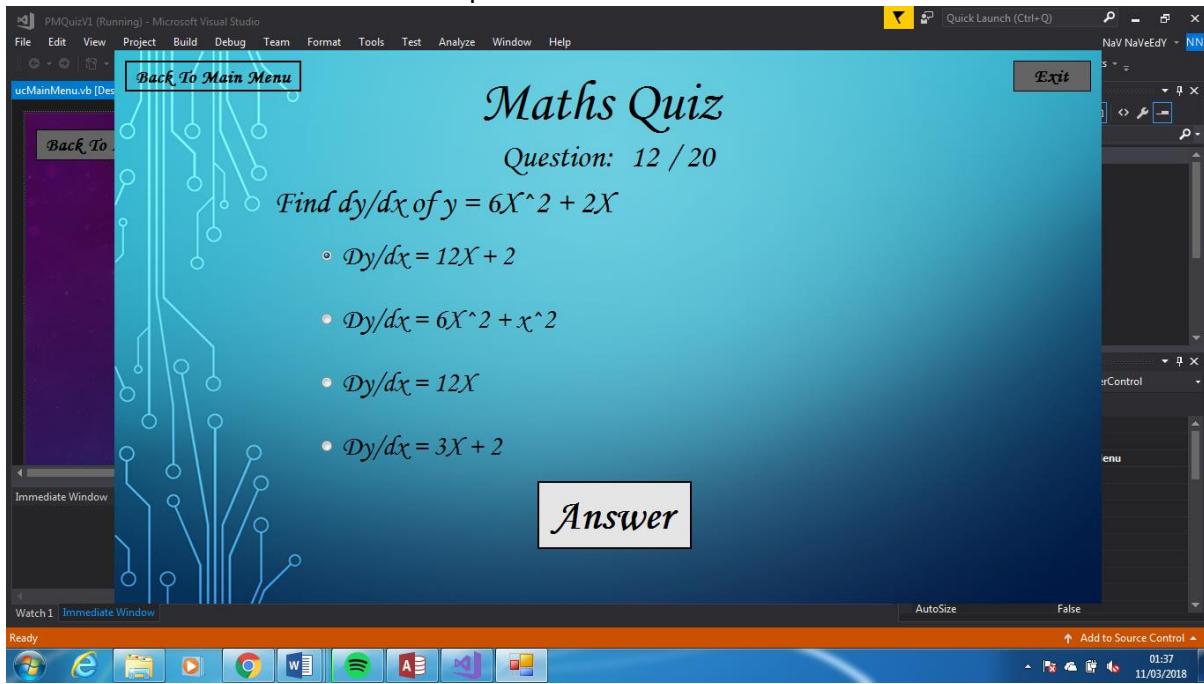
Candidate Number: 1904



**Test 14:**

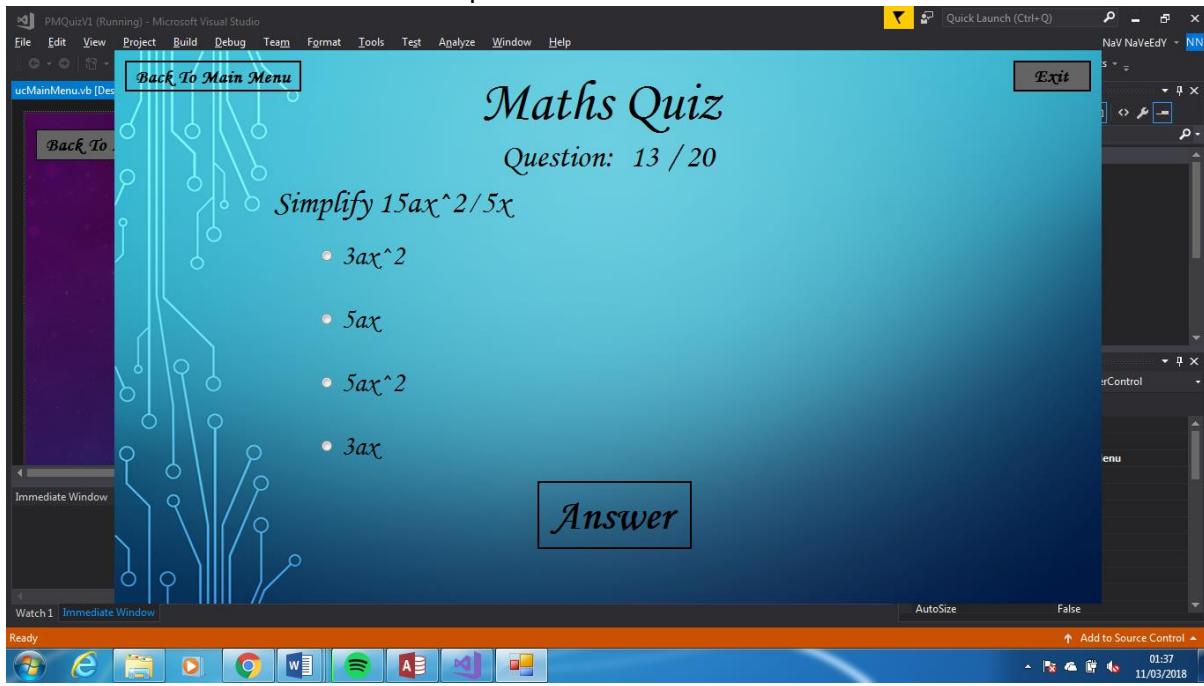
Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

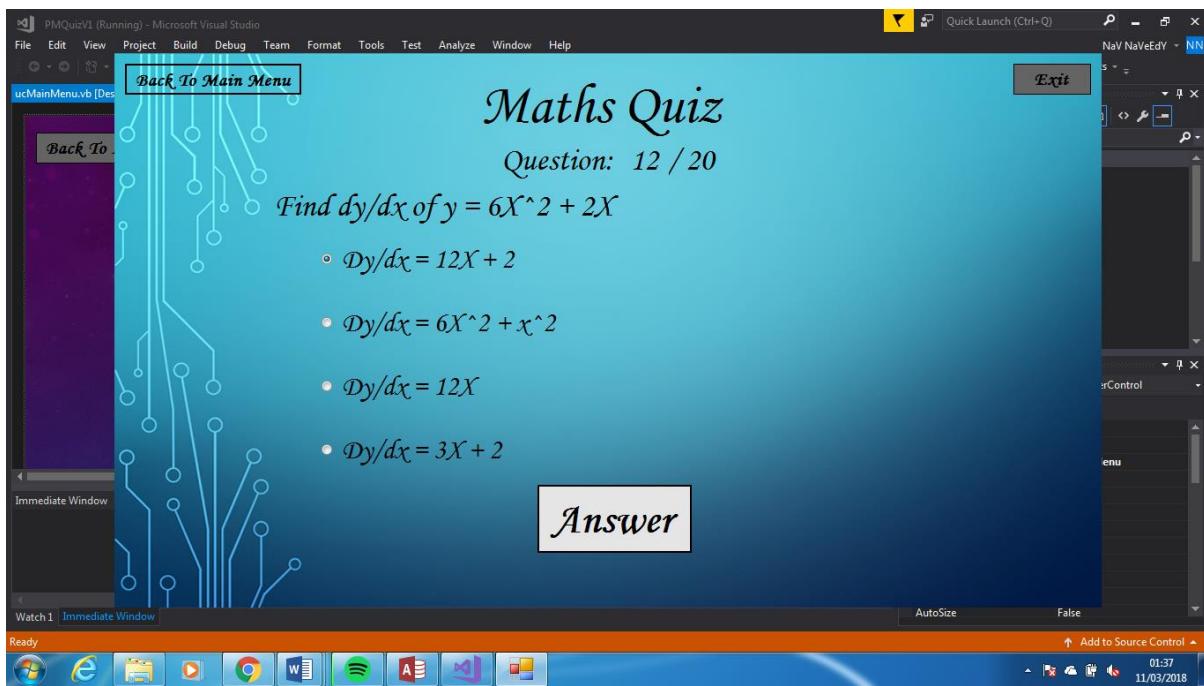


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

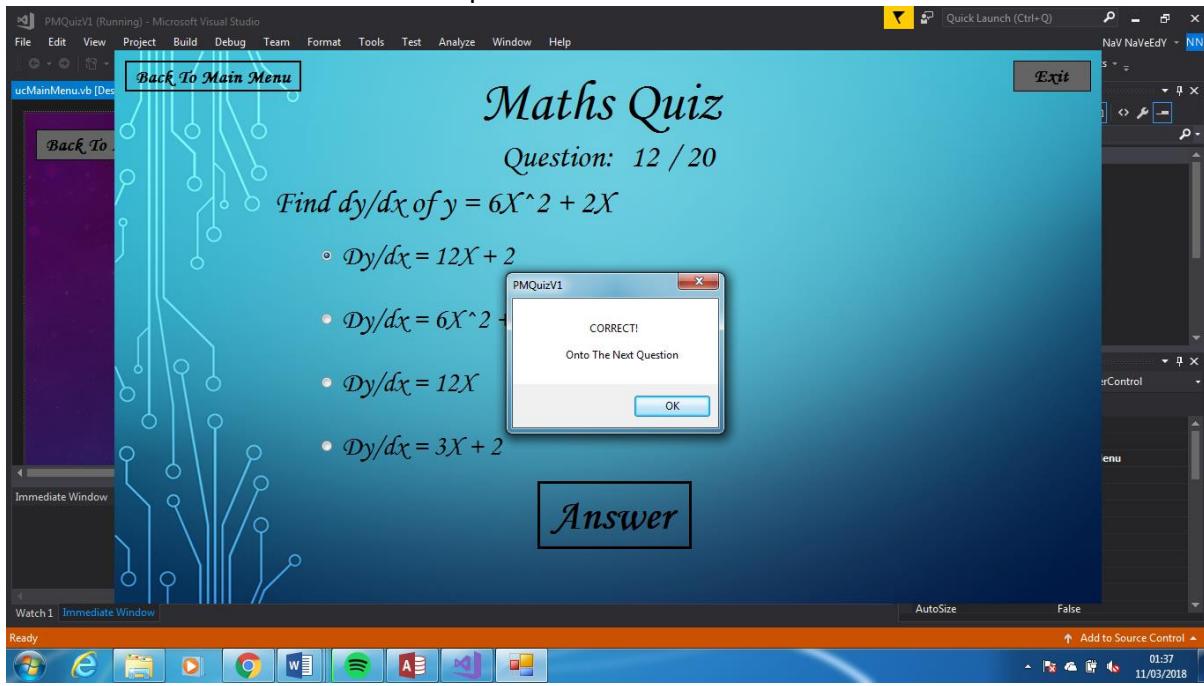


### Test 15:

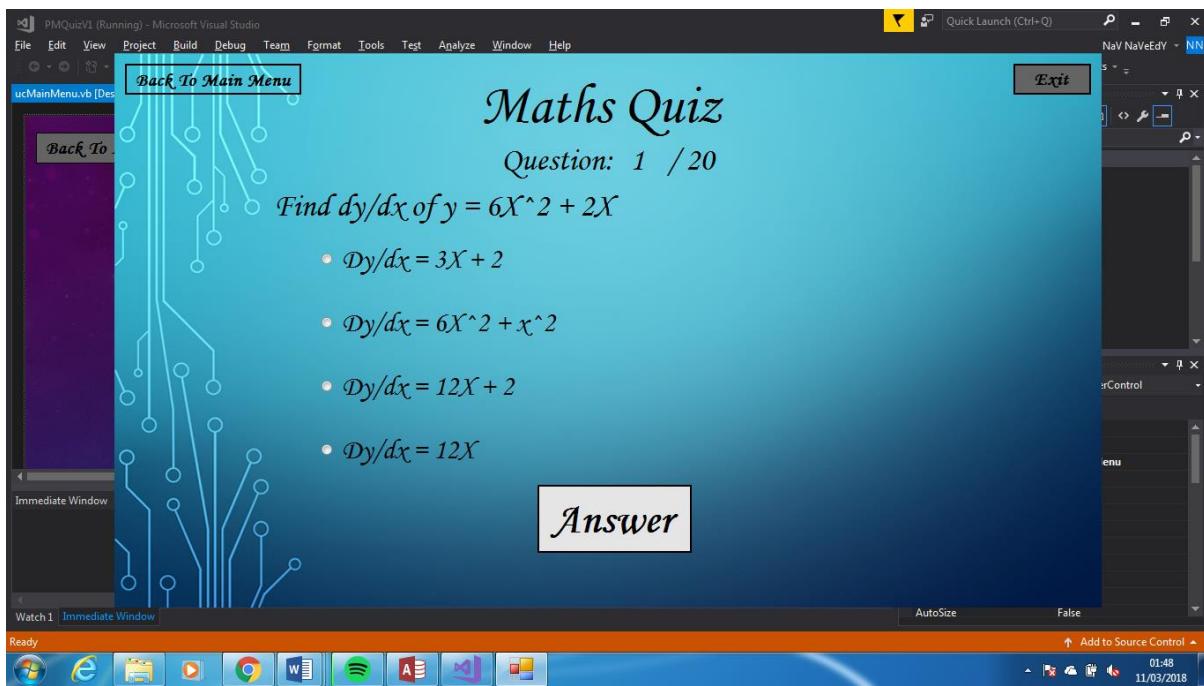


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

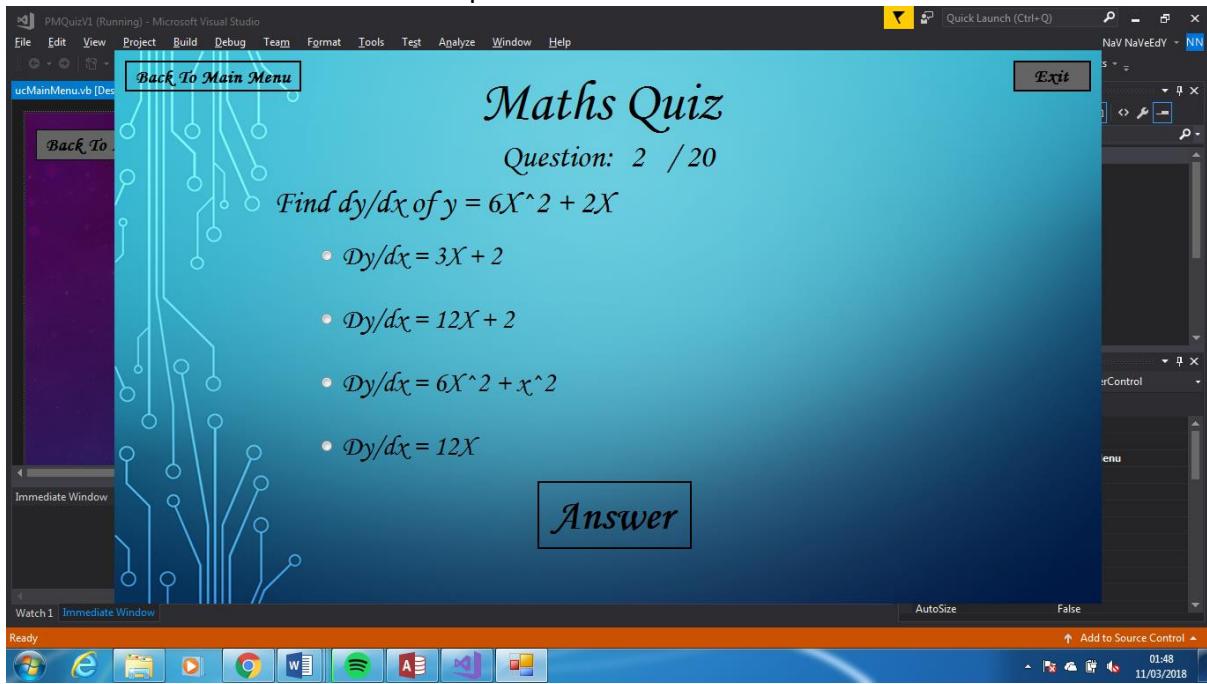


### Test 16:

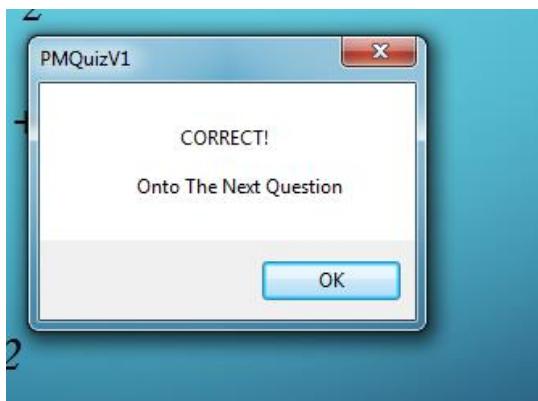


Candidate Name: Naveed Ali Rafeeq

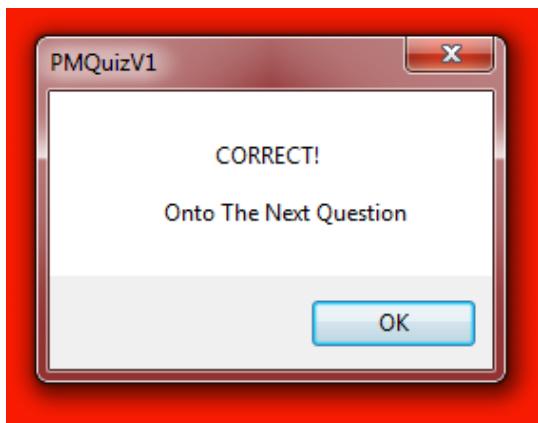
Candidate Number: 1904



### Test 17:



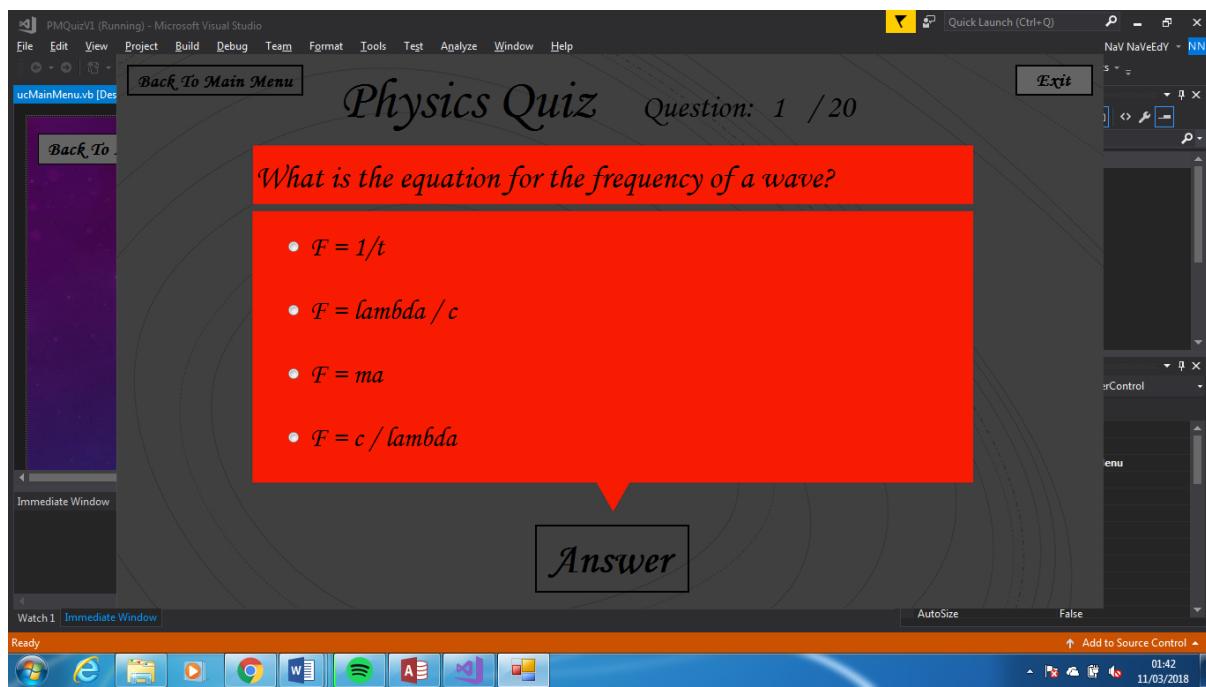
### Test 18:



Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

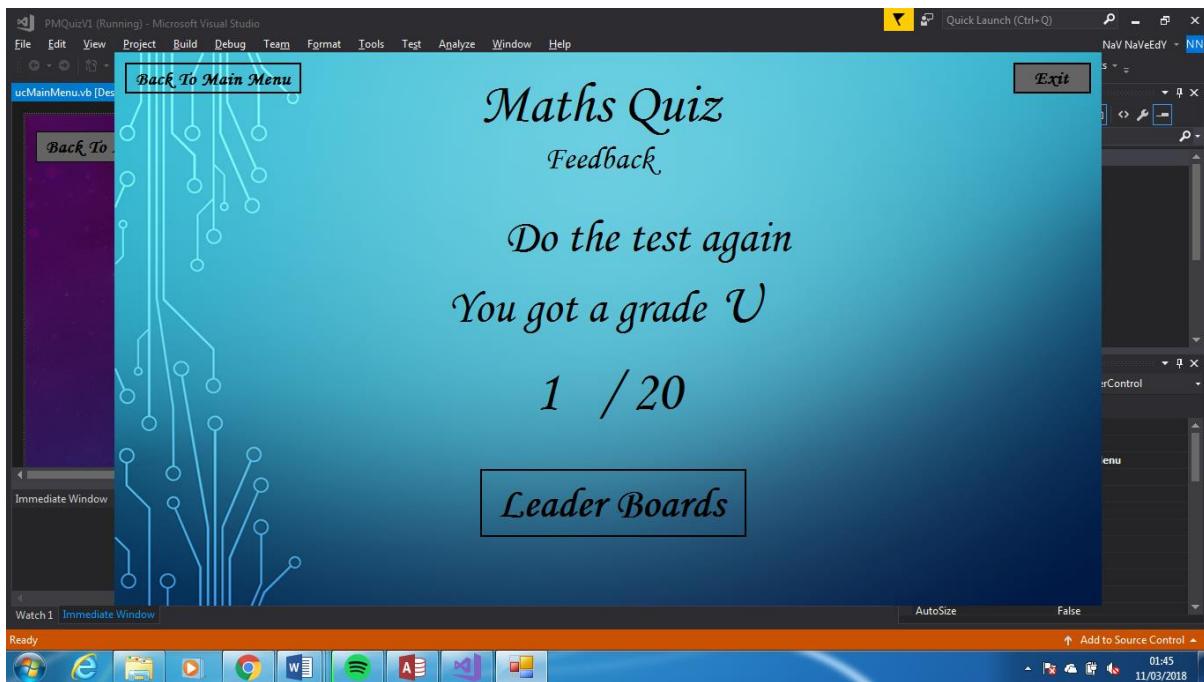
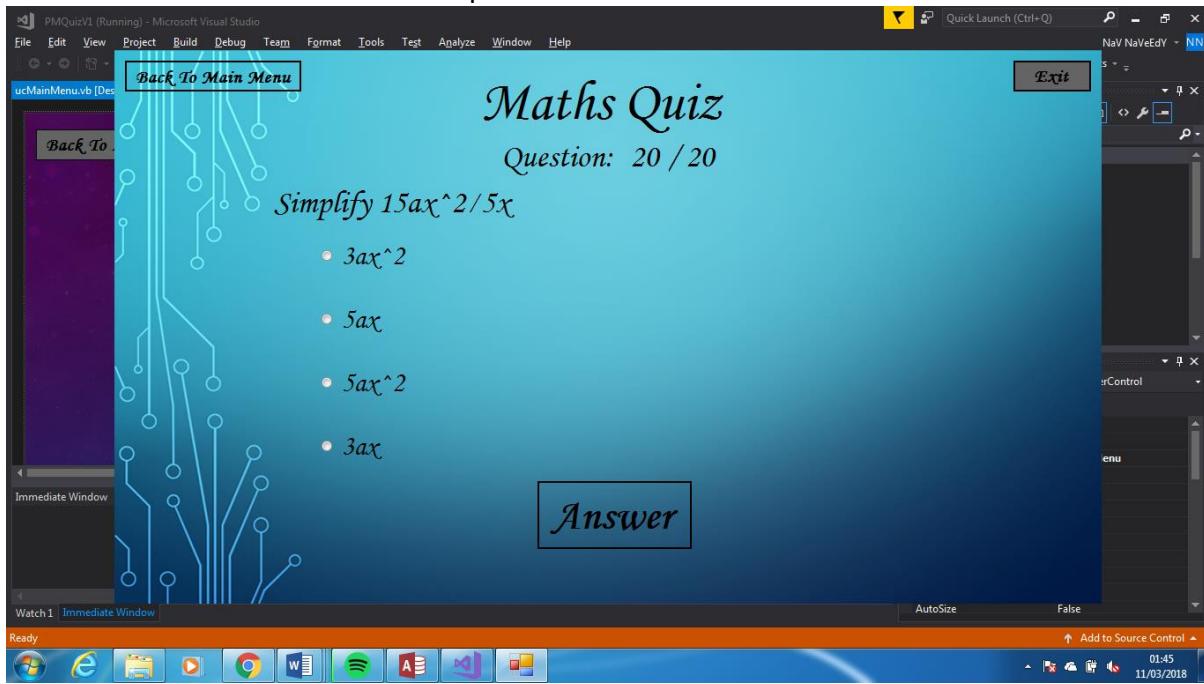
**Test 19:**



**Test 20:**

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



**Test 21:**

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

**MQuestion Table Data:**

	MA1	MA2	MA3	MA4
Find $dy/dx$ of $y = 3X^2 + X$	$Dy/dx = 6X + 1$	$Dy/dx = 6X$	$Dy/dx = 3X + 1$	$Dy/dx = 6X^2 + x$
Find $dy/dx$ of $y = 6X^2 + 2X$	$Dy/dx = 12X + 2$	$Dy/dx = 12X$	$Dy/dx = 3X + 2$	$Dy/dx = 6X^2 + x$
Simplify $15ax^2/5x$	$3ax$	$3ax^2$	$5ax^2$	$5ax$
What is used to show differentiation of the equation $y = f(x)$ ?	$dy/dx$	$dx/dy$	$y/x$	$dy/df$

### Test 22:

**MQuestion Table Data (Row 2 Selected):**

	MA1	MA2	MA3	MA4
Find $dy/dx$ of $y = 6X^2 + 2X$	$Dy/dx = 12X + 2$	$Dy/dx = 12X$	$Dy/dx = 3X + 2$	$Dy/dx = 6X^2 + x$
Simplify $15ax^2/5x$	$3ax$	$3ax^2$	$5ax^2$	$5ax$
What is used to show differentiation of the equation $y = f(x)$ ?	$dy/dx$	$dx/dy$	$y/x$	$dy/df$

**Maths Quiz Application Screenshot:**

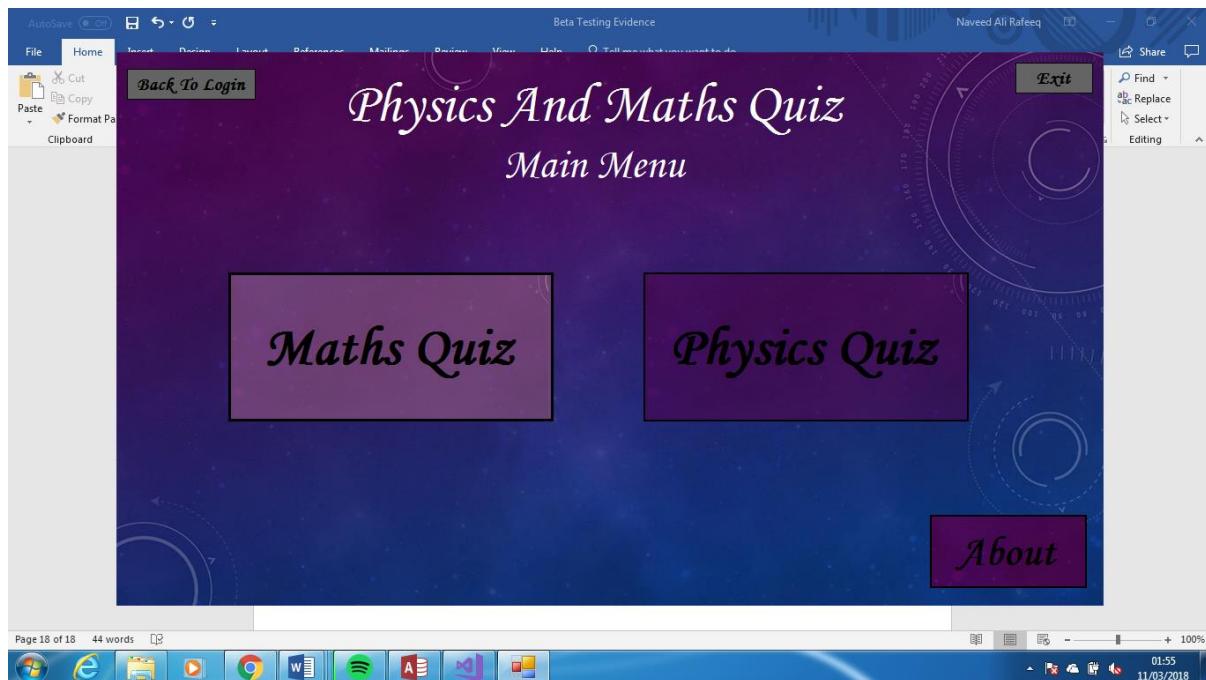
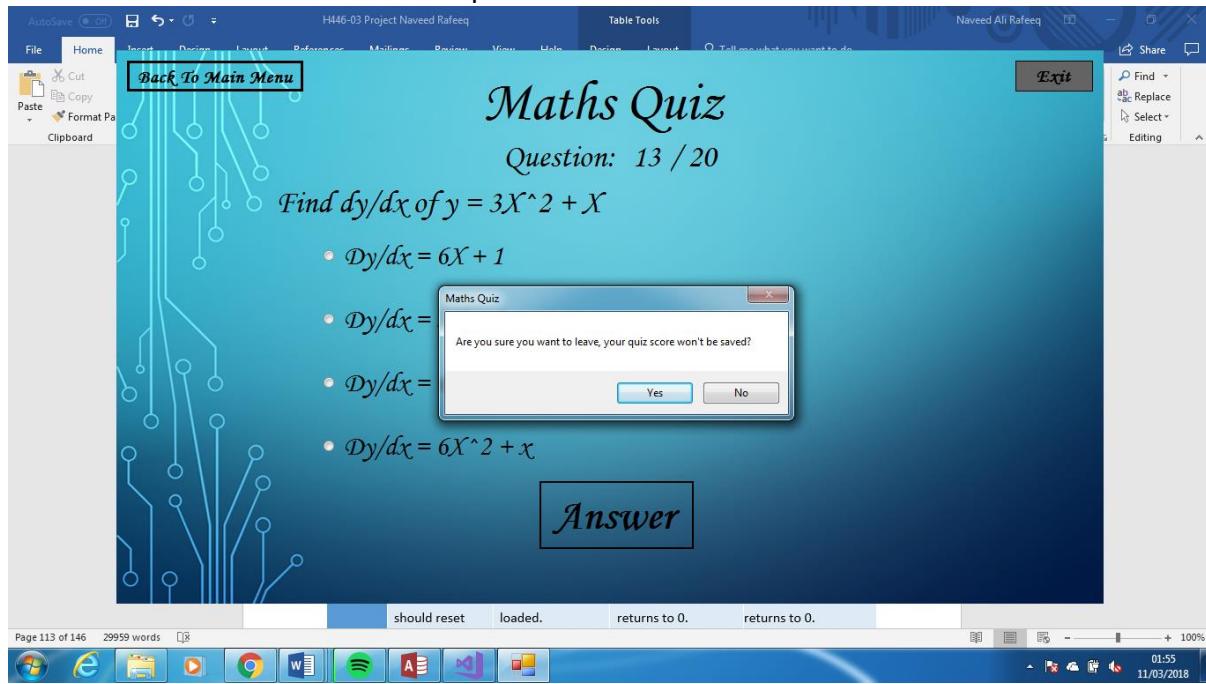
Question: 2 / 20

Find  $dy/dx$  of  $y = 6X^2 + 2X$

- $Dy/dx = 3X + 2$
- $Dy/dx = 12X + 2$
- $Dy/dx = 6X^2 + x^2$
- $Dy/dx = 12X$

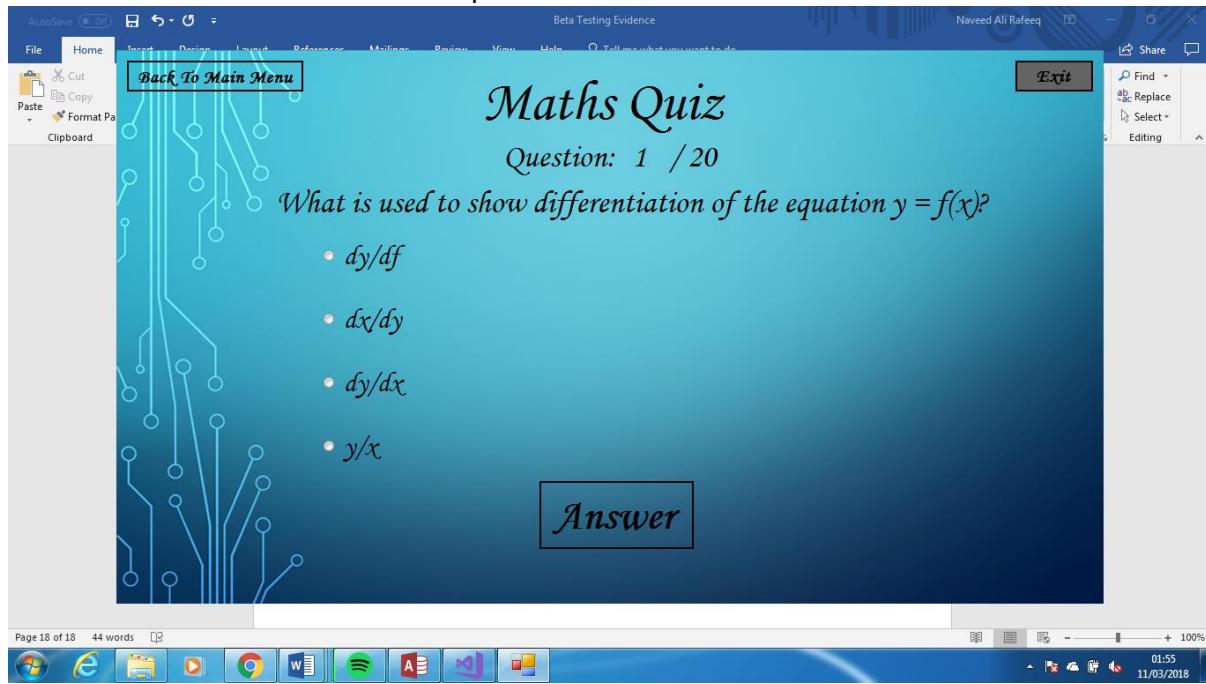
**Answer**

### Test 23:

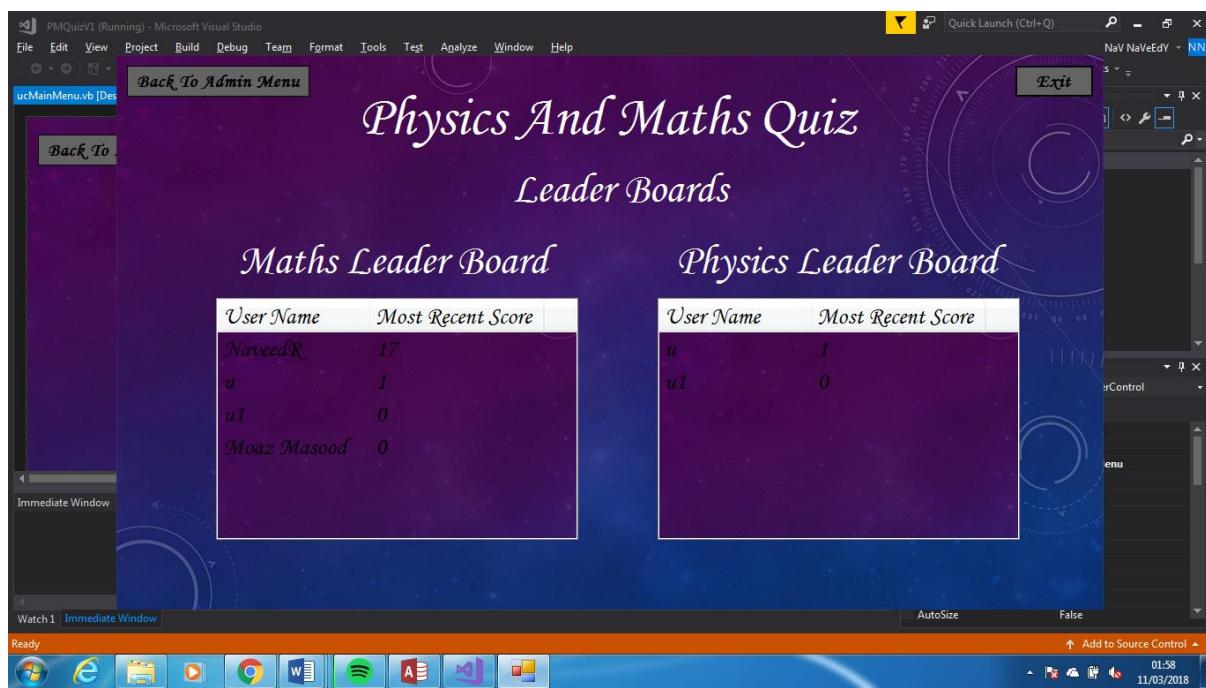


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



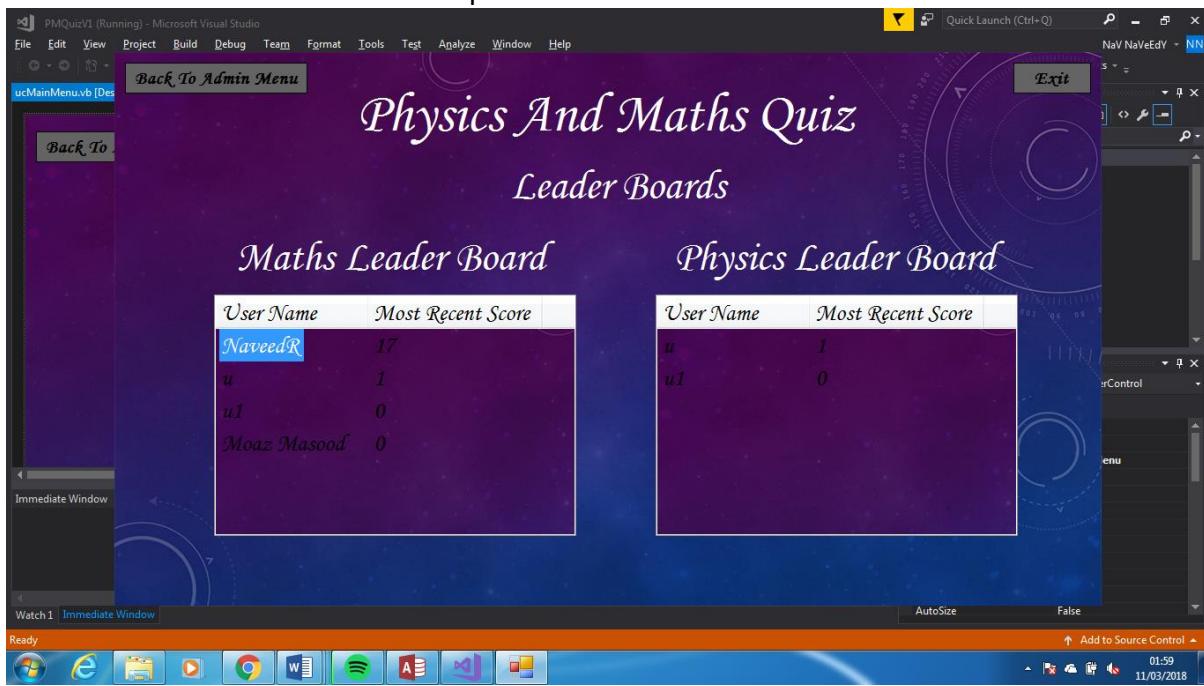
### Test 26:



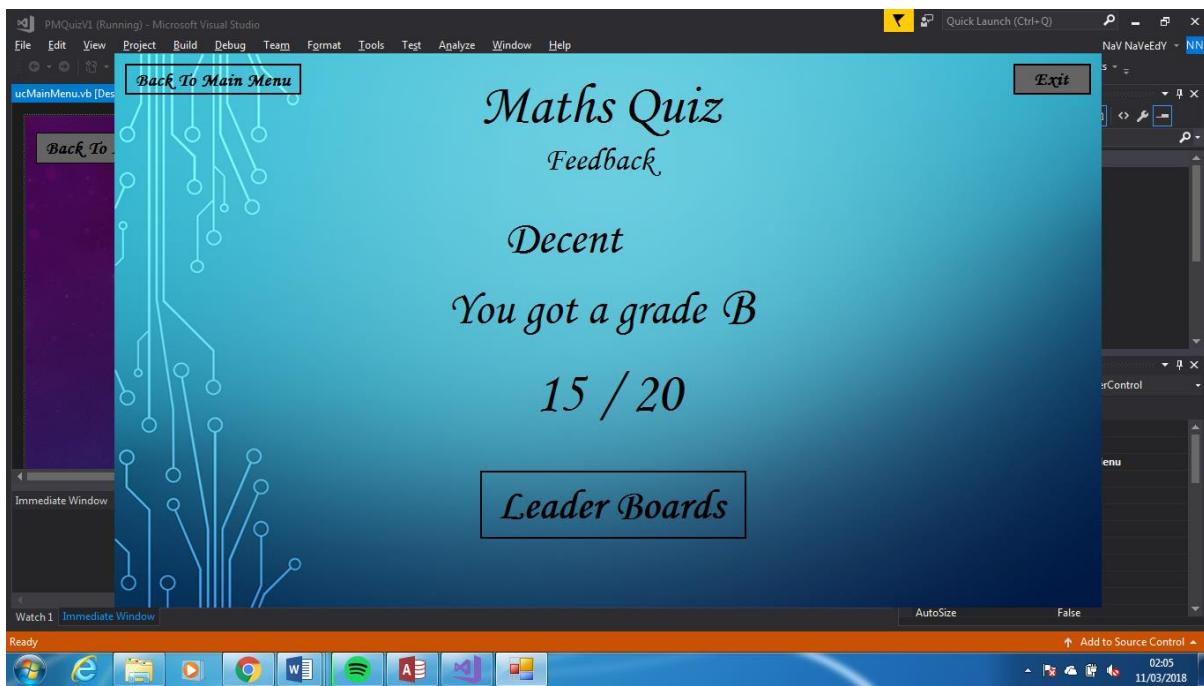
### Test 27:

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



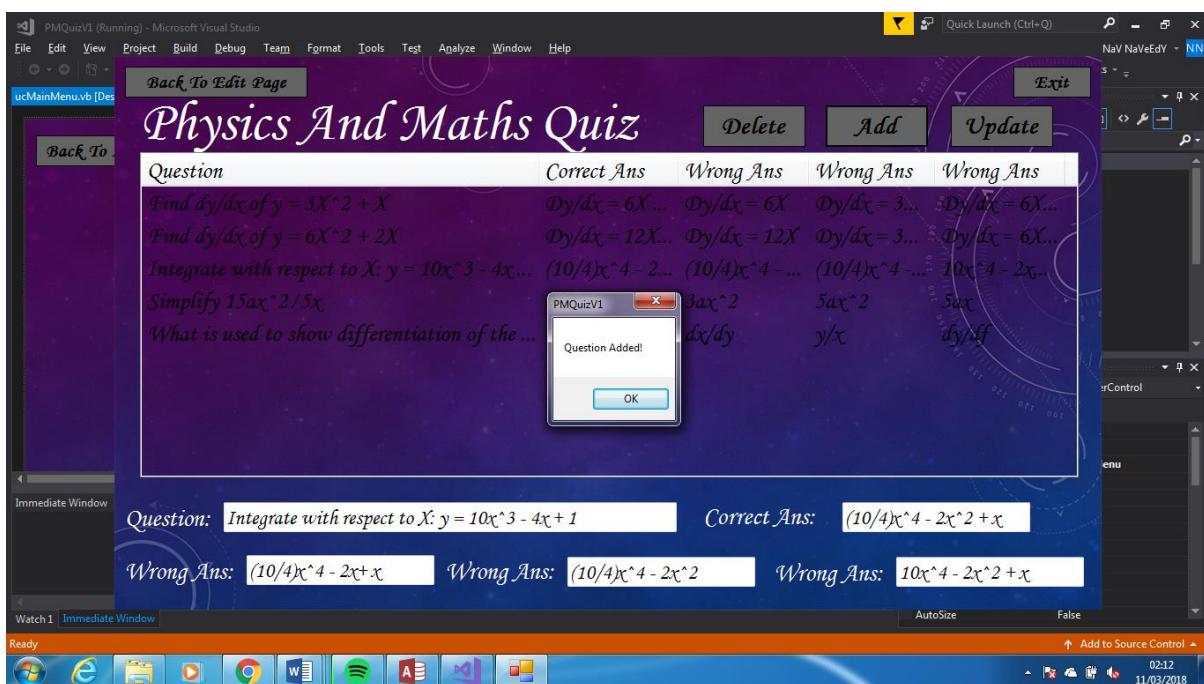
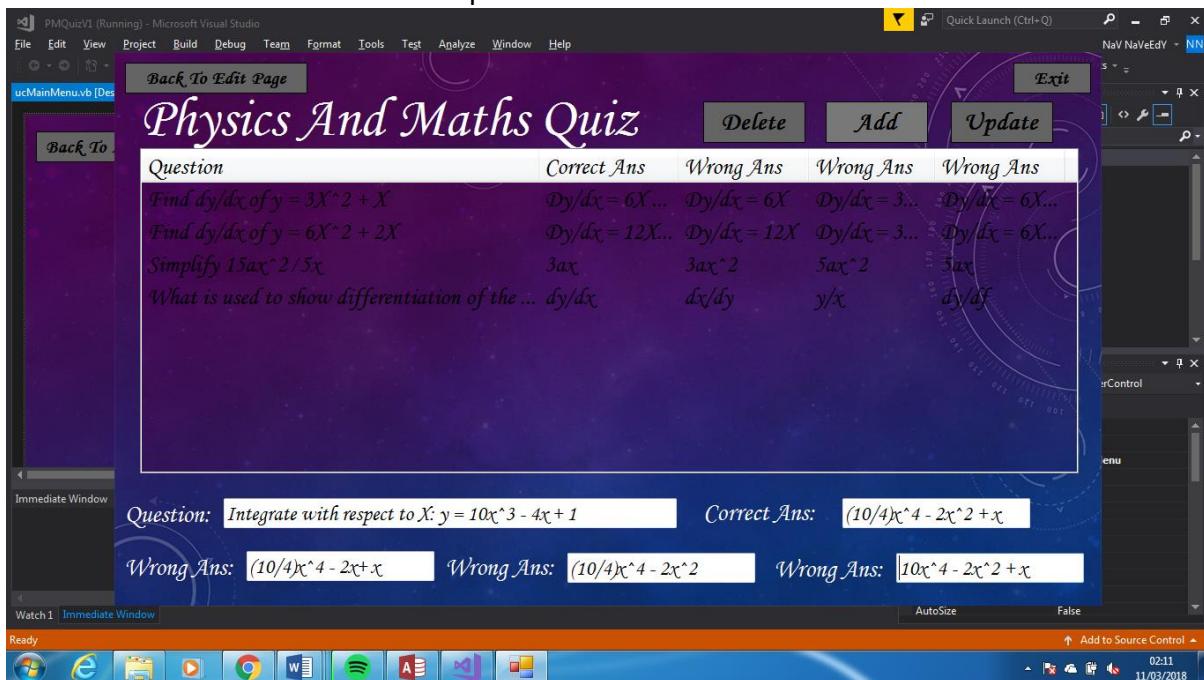
### Test 28:



### Test 29:

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

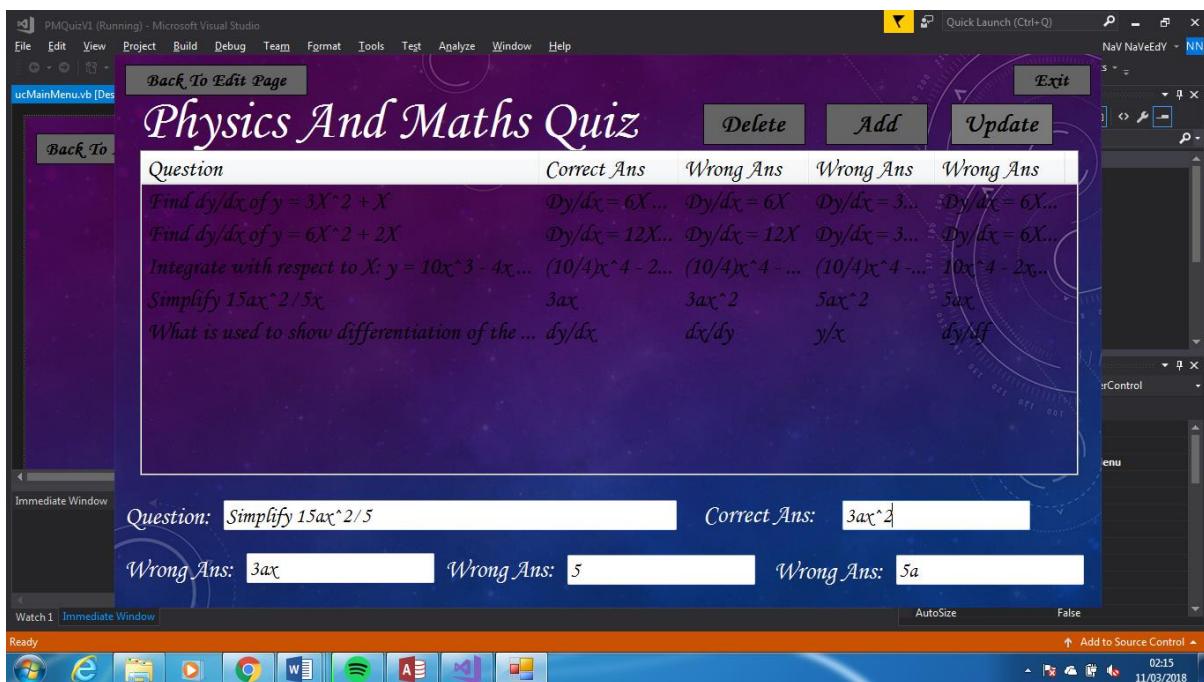
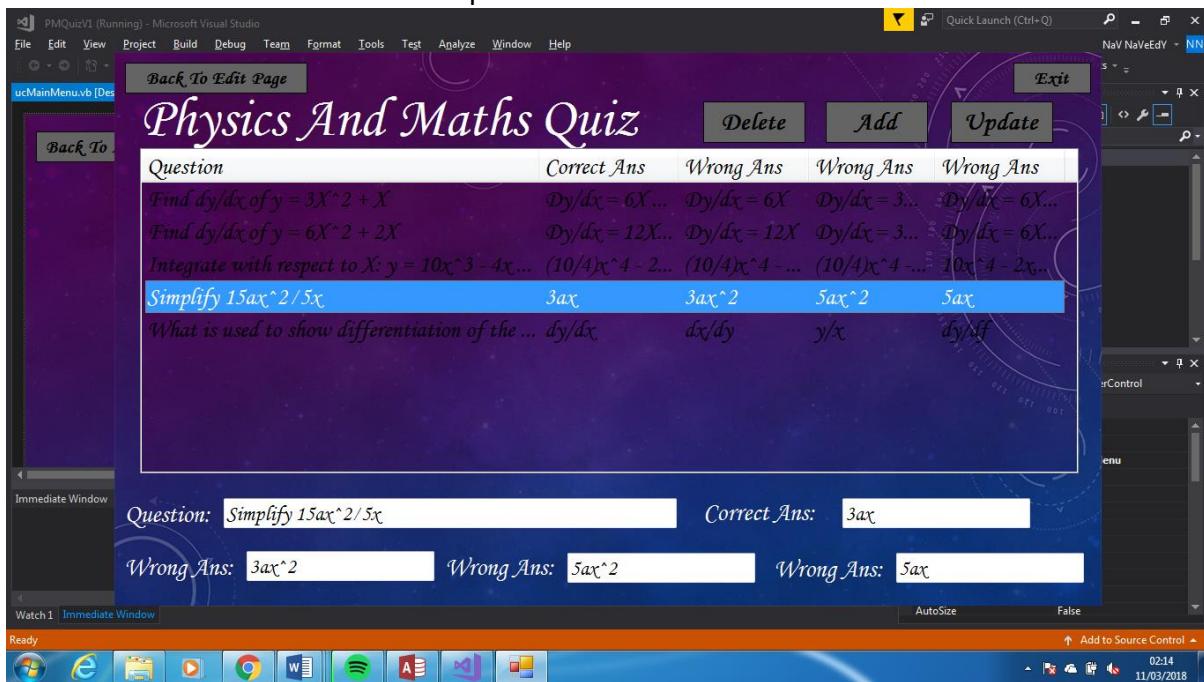


MQuestion	MA1	MA2	MA3	MA4	Click to Add
Find $dy/dx$ of $y = 3x^2 + x$	$Dy/dx = 6x + 1$	$Dy/dx = 6x$	$Dy/dx = 3x + 1$	$Dy/dx = 6x^2 + x$	
Find $dy/dx$ of $y = 6x^2 + 2x$	$Dy/dx = 12x + 2$	$Dy/dx = 12x$	$Dy/dx = 3x + 2$	$Dy/dx = 6x^2 + x$	
Integrate with respect to $x$ : $y = 10x^3 - 4x + 1$	$(10/4)x^4 - 2x^2 + x$	$(10/4)x^4 - 2x^2$	$(10/4)x^4 - 2x^2$	$10x^4 - 2x^2 + x$	
Simplify $15ax^2/5x$	$3ax$	$3ax^2$	$5ax^2$	$5ax$	
What is used to show differentiation of the equation $y = f(x)$ ?	$dy/dx$	$dx/dy$	$y/x$	$dy/df$	

### Test 30:

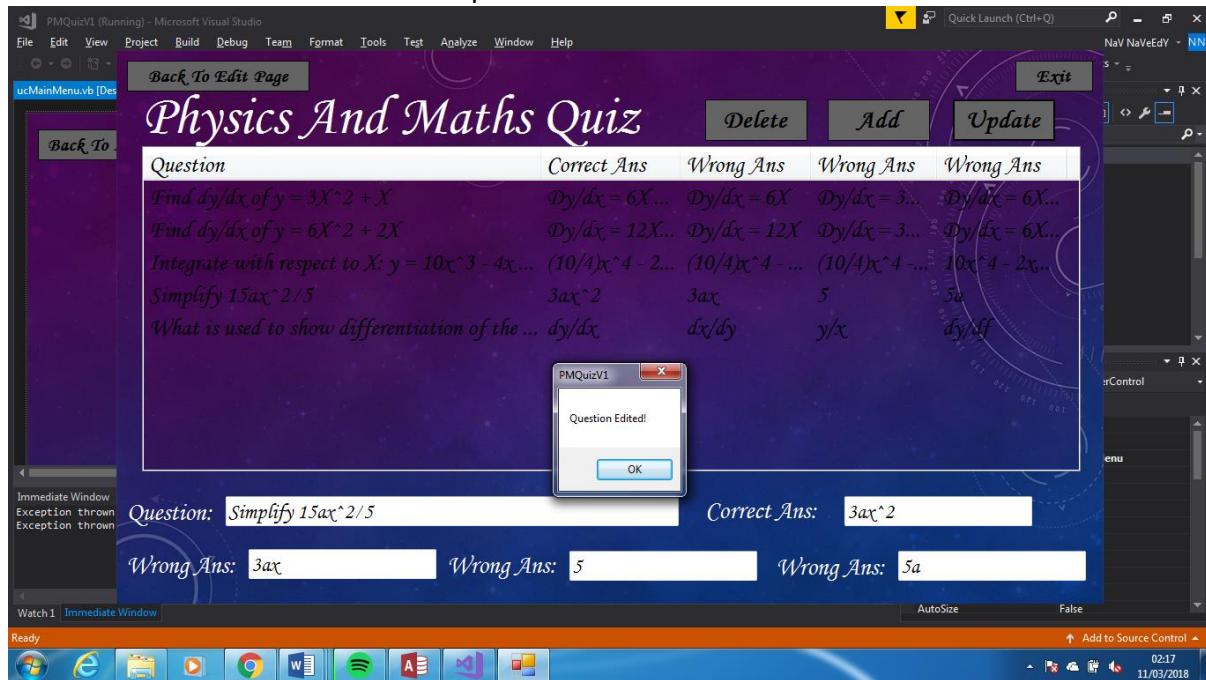
Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

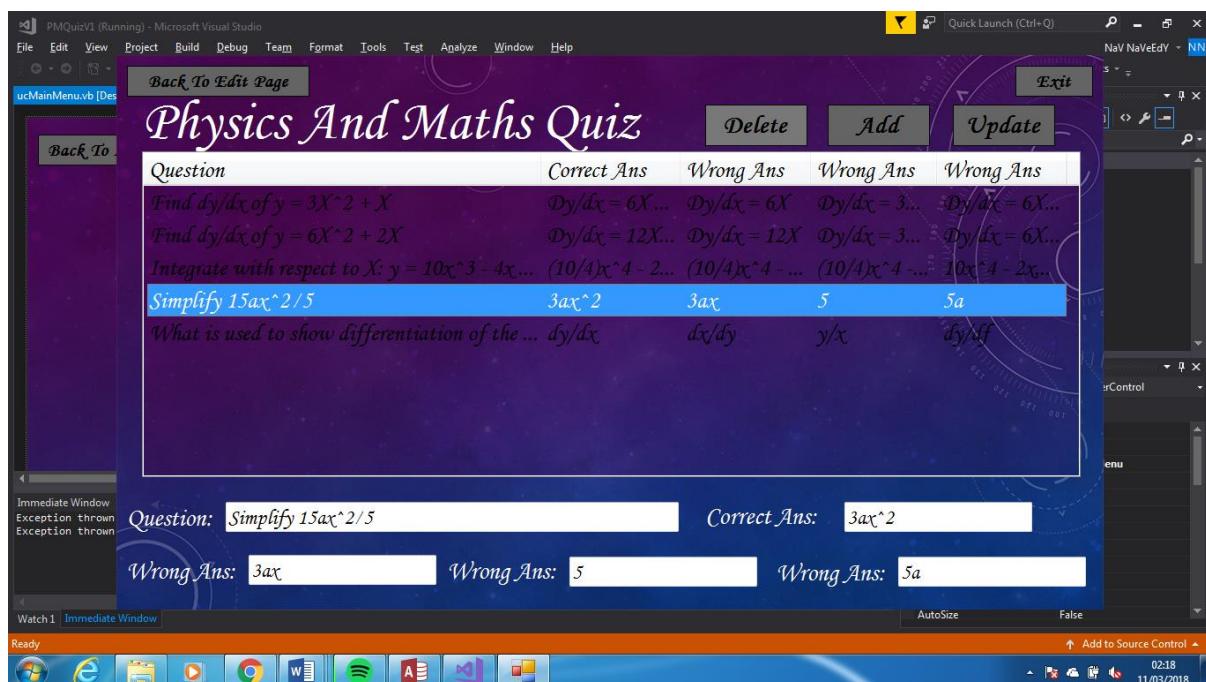


Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

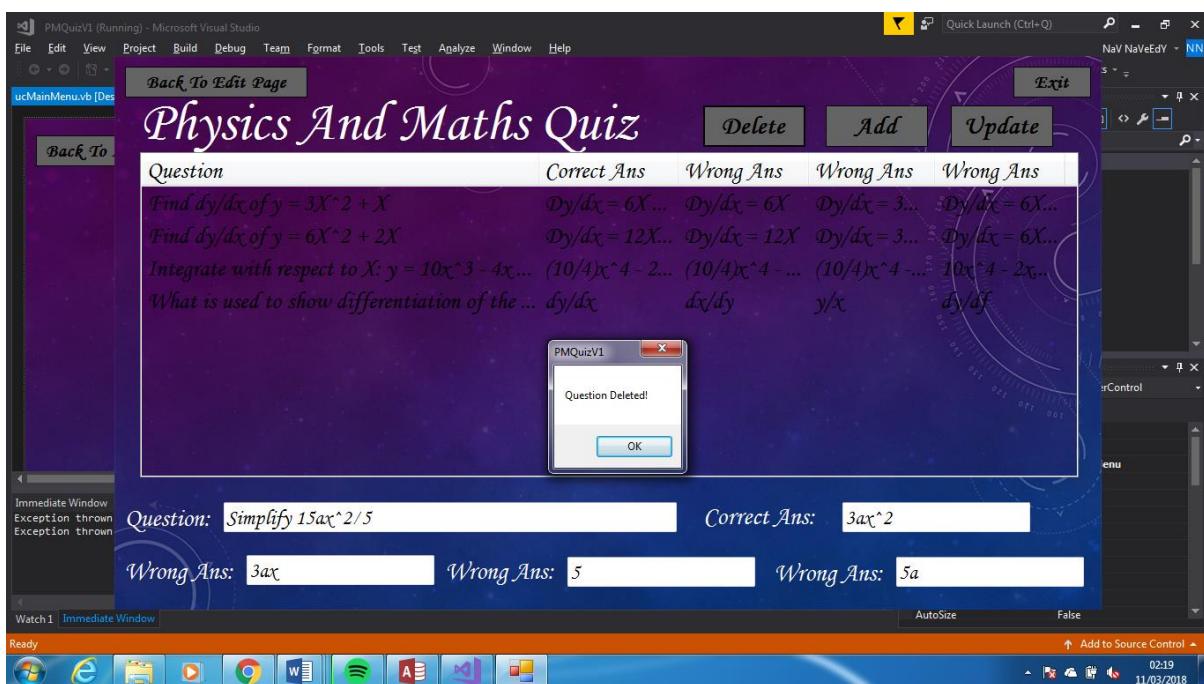
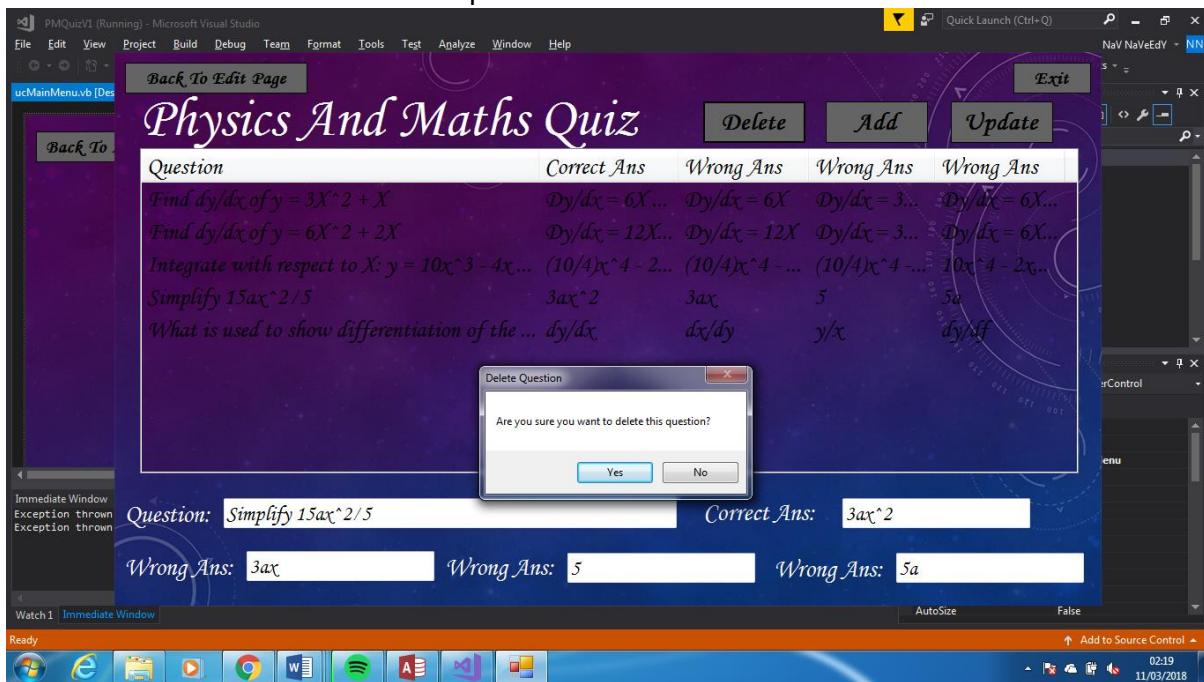


### Test 31:



Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904



MQuestion	MA1	MA2	MA3	MA4	Click to Add
Find dy/dx of y = 3X^2 + X	Dy/dx = 6X + 1	Dy/dx = 6X	Dy/dx = 3X + 1	Dy/dx = 6X^2 + x	
Find dy/dx of y = 6X^2 + 2X	Dy/dx = 12X + 2	Dy/dx = 12X	Dy/dx = 3X + 2	Dy/dx = 6X^2 + x	
Integrate with respect to X: y = 10x^3 - 4x + 1	(10/4)x^4 - 2x^2 + x	(10/4)x^4 - 2x + x	(10/4)x^4 - 2x^2	10x^4 - 2x^2 + x	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	
What is used to show differentiation of the equation y = f(x)?	dy/dx	dx/dy	y/x	dy/df	
*					

## Evaluation Of Solution:

### Success Criteria:

**User/Staff Login before loading up the quiz menu:** Requirement has been met since the user has to login with correct details before accessing the main menu. (Test No 2)

**Quiz Menu:** Requirement met since my program has a menu that allows you to choose which subject quiz you wish to do. (Test No 11)

**Scoring System:** Requirement met, program contains a counter that keeps track of how many questions you got right and then tells you after. (Test No 28)

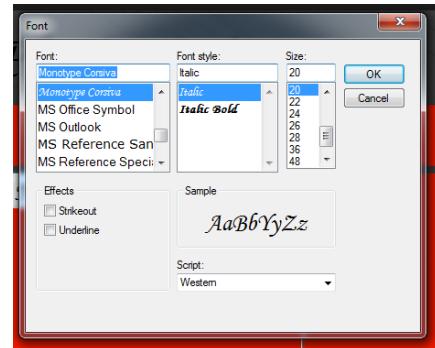
**Exit Button:** Requirement met, top right of each UC is the Exit button that closes the program. (Test No 8)

**The quiz is navigated with the mouse:** Requirement met, full of buttons that all need LMC to trigger them.

**A menu with 5 options:** Requirement met, 3 main buttons in the middle and one in each top corner. (Test No 8, 9 & 11)

**The font size is above 14:** Requirement met, smallest font size = 20.

**There are 20 questions in each quiz:** Requirement met, each quiz contains 20 questions and then loads the Feedback UC. (Testing No 20)



**There must be more than one topic included within the quiz:** Requirement not met, currently only one topic in a quiz. I did not meet this requirement due to the fact that I did not have enough time, I only managed to create the basics which was the Maths and Physics Question tables I started the program and design so that there is only one table for Maths questions and another for Physics, however I realized to implement more than one topic within a quiz I had to group each topic questions into different tables. By doing this I could have had a random number generating function with each number assigned to each table and hence each topic.

**A Message Box appearing after answering each question:** Requirement met, after choosing a radio button and clicking the answer button a message box would appear telling the user whether they are correct or incorrect. (Test No 15, 17 & 18)

**Radio Buttons:** Requirement met, only one question at a time can be checked. (Test No 15)

**When the user finishes the quiz, a display should appear telling them what score they got out of 20:** Requirement met, the Feedback UC gives the user their score out of 20 and a message as well. (Test No 28)

**After closing the message box, a button appears to allow the user to go to the next question:** Not met, I decided not to implement this button into my development because I realised this would become quite a tedious button since the user is obviously going to continue to the next question. Therefore, I made it so that when the message box closes the User Control Refreshes and the new question loads automatically, which makes the program much more efficient.

**Menu should be easy to use:** Requirement met, the large buttons that highlight when the mouse hovers over them makes it quite easy to use. My stakeholders were impressed by the simplicity of the appearance of the Program which made it very easy to use.

**A title should be displayed on the Menu:** Requirement met, not only is a title displayed on the Menu but there is one displayed on every User Control.

**Appeal to ages 16+:** Requirement met due to the questions being part of AS Maths and Physics specifications, which is usually taught after GCSE's at the ages of 16 and higher.

**Should display a message at the end of the quiz next to your score:** Requirement met, a bespoke message is written depending on your grade which depends on your score which is displayed in the Feedback UC after the quiz had been completed. (Test No 28)

**A grade will be displayed at the end of the quiz:** Requirement met, I added the select statement that assigns a grade and message depending on what score the user gets. (Test No 28)

**The teacher's will have an extra 2 options, Maths and Physics leader boards:** Requirement partially met, the teachers are not able to take quizzes or access the about screen. I did this because the teachers are the users who create all of the questions and answers therefore there is no point of them undergoing the quizzes since they know all of the answers. I did however surpass this success criteria, I managed to do this by creating an Edit Page. This Edit Questions User Control gave the teacher full control over their students, they could delete their students accounts (only the ones in their class), they could also add, delete and edit any question and answer within the Maths and Physics Question tables. This therefore gave their admin accounts a full purpose and not just so that they do not have to do the quiz before they could access the leader boards. (Test No 29, 30 & 31)

**A sort algorithm to arrange the order of the leader boards:** Requirement not met, I did not need to use a sort algorithm to arrange the order of the leader boards since my SQL statements did it for me when loading data into the Data Tables. (Test No 26)

## Usability Features:

- Many parts of the program I created a sub that allows a button to be clicked by pressing the enter button within the textbox. Examples of this were in the Login UC when in the password textbox and the Create Account UC when in the TID textbox.

```
Public Class ucLogin
    Dim ucLoad As MainForm

    'Does a check to see if a key has been pressed when the user is in the password textbox
    Private Sub PassTxt_PreviewKeyDown(sender As Object, e As PreviewKeyDownEventArgs) Handles PassTxt.PreviewKeyDown
        If e.KeyCode = Keys.Enter Then
            Call LoginBtn_Click(sender, e)
        End If 'When the user presses enter button in the password textbox it runs the login button
    End Sub
```

However, I should have implemented this into all of my textboxes which therefore would allow the user to use the Program with more ease. Therefore, I only partially met this usability feature, if I had more time I would have been able to add this sub routine to every textbox.

- Tabbing from button to button, textbox to textbox and button to textbox was also part of my program; you could see which button was selected from tabbing by it having a dark ring surrounding that button and you could see which textbox has been selected by the flashing writing cursor displaying within the textbox. This helps due to the fact that all that was needed to click the button was to press the enter button. I would say this usability feature is only partially met as well since I could have improved this element of my program by controlling the order in which the tabbing occurs which would improve the time taken to get to the desired button of choice.
- During the actual quiz sections of the program, the number at the top would inform the user which question they are on out of 20. This was a very useful usability feature since the user would never need to remember what question they are on and how many questions are left. But, I did not supply the user how many questions they currently have got right out of 20 whilst the quiz was running. It would obviously inform the user during the feedback User Control but not during the quiz itself. Therefore, I could have improved this usability feature. I would say that I met this usability feature because I still informed the user how many they got right in the feedback after the quiz, which is effectively the most important part of having a right answer counter. The fact that it is not there during the quiz will probably go unnoticed. (Test No 17 & 28)

- The simple design of the program was another usability feature that was implemented. Due to the simple design the user could easily navigate themselves through the Quiz with minimum effort. The only time however when things got a bit complicated was in the Admin Main Menu where the teacher could edit, delete and add questions. I could have separated them into their own User Controls to keep with the simplicity of the Quiz. Therefore this was another usability feature that I met. (All the Test Evidence should suffice)
- Another usability feature I added was the buttons highlighting when the mouse would hover over them. This was particularly useful since some of the buttons had the same design as the background therefore when they highlighted the user knew that it could be selected. The buttons with the grey background highlighted automatically but the custom background buttons I had to create the opaque background to match it and display when the mouse hovered over it, therefore, meeting this usability feature. But, there was sometimes a lag delay between hovering over the button and the opaque background appearing which was quite frustrating and was a limitation within the whole project itself as seen below. (Test No 11, 13 & 23)
- The fact that the user could edit the sizes of the leader board columns to make them more bespoke was also another usability feature. This was useful since some usernames might have been too large for the column width and therefore the column would need adjustment. I would say I met this usability feature because not only does the column width increase and decrease but the columns also increase as well, and you can scroll through all of the usernames and scores. (Development Testing Prototype 3: No 6 and 9)

### Maintenance & Limitations:

There are many problems and bugs within my program that would either need to be fixed if I carried on developing my Program. There are also some limitations to my Program, this was either due to my lack of knowledge of programming or due to Visual Basic itself.

- My main problem with my Quiz is the fact that it lags. Every time I load different User Controls there is always lag to it, and the components will load at different times. When some User controls load it is not that bad, like the Main Menus or the Login UC. Others it takes a few seconds of buffering till all the components load like the Edit Questions UC or the Quiz UCs. Even though I do not know how to prevent the whole program from lagging, I could have tried better to minimise it. A lot of my User Controls contained too many items within them which therefore caused the lag, if I had created more User Controls all containing smaller amounts the Quiz would have lagged less. An example of where I could have done this was in the Edit Questions UC. Instead of having 3 separate

list views as well as their own components I could have done one of the two. I could have created 3 more UCs that each contained their own separate list views and buttons this way it would have lagged less when loading the Edit Question UC and would lag less when loading each separate UC. Alternatively, I could have had one list view and its corresponding buttons and just loaded each of the tables into the one list view. I could have used a select case statement to decide which data table to fill it with as well as position co-ordinates and size co-ordinates to resize the list view to accommodate for the different tables.

- Another bug I had was the fact that the users could just go onto the next question without clicking a radio button. This meant that no validation was implemented within the quiz User Controls. If I had more time I would have created an If statement similar to validation If statements used in the Create Account UC. If one of the radio buttons was not selected, then it would have exited the sub that runs when the answer button is clicked. Thus, making it useless if no radio button is checked.
- Another problem with my program was the fact that it would allow the admins to add blank questions. This is a huge problem since if that blank question then comes up as a question that a student has to answer it would be unfair since it would literally be up to chance whether the radio button he selected would be the blank placed in the first answer column. If I had the time I would have prevented this by creating an If statement like the one within the Create Account UC where if any of the textboxes are blank the sub that runs when the Add button is clicked will be exited.
- Another bug is the fact that admins can add the same question in as many times as they want within the Question tables. This is a bug since it is not needed due to the fact that it just makes the users more likely to answer to same question. Meaning that the users will not be able to answer a range of questions that can help them improve their knowledge. I could have prevented this by adding a form of verification in the form of an If statement. I would have placed the contents of the sub that runs when the Add button is clicked into an If statement and would only run if a Data Table contains no rows since it would look for the question that is contained within the textbox. A bit like the one used within the Create Account UC where it checks if the username is already in use.
- There is one huge limitation with my program that makes it very impractical. The fact that my database is only saved in the same folder as the executable file makes the Quiz very limited. The reason I did this was due to the fact that it made writing to and reading from the Access Database much easier since it was in the same folder. For the Quiz to work to its full potential I would need to make the database be part of a cloud storage system so that the data can be accessed by the teachers and students anywhere through the program. If I was to do this it would make sense to create a website in

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

HTML, JavaScript and CSS in order to sustain such a program online so that users would not need to download the program instead it would all be stored online. However, the reason I never used an online database was due to the fact that I have had no experience in that field of database programming and so due to my current coding knowledge I would not know how to write, read and edit information to such complex databases.

## PROJECT APPENDIXES

**Public Class Details** 'These variables are now global therefore, when the uc's is disposed the variable data is not taken with it

```

Public Shared QtnRight, QtnTotal As Integer
Public Shared LoginUser, UserTID As String

End Class

Public Class MainForm

    Private Sub MainForm_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        addUC("ucLogin")
        'Places the User Control Login when the mainform loads
    End Sub

    'Sub routine which takes in user controls, and makes a choice on which uc to load
    Public Sub addUC(ByVal ucChoice As String)
        Dim newUC As UserControl 'variable newUC that adds new user control
        newUC = New ucLogin 'Assigns a User Control to newUC so the variable isn't empty

        Select Case ucChoice
            Case Is = "ucLogin"
                newUC = New ucLogin 'Loads the user control ucLogin
            Case Is = "ucMainMenu"
                newUC = New ucMainMenu 'Loads the user control ucMainMenu
            Case Is = "ucCreateAcc"
                newUC = New CreateAccBtn 'Loads the user control ucCreateAcc
            Case Is = "ucAbout"
                newUC = New ucAbout 'Loads the user control ucAbout
            Case Is = "ucMathsQuiz"
                newUC = New ucMathsQuiz 'Loads the user control ucMathsQuiz
            Case Is = "ucPhysicsQuiz"
                newUC = New ucPhysicsQuiz 'Loads the user control ucPhysicsQuiz
            Case Is = "ucMathsFeedback"
                newUC = New ucMathsFeedback 'Loads the user control ucMathsFeedback
            Case Is = "ucPhysicsFeedback"
                newUC = New ucPhysicsFeedback 'Loads the user control ucPhysicsFeedback
            Case Is = "ucAdminMainMenu"
                newUC = New ucAdminMainMenu 'Loads the user control ucAdminMainMenu
            Case Is = "ucLeaderBoards"
                newUC = New ucLeaderBoards 'Loads the user control ucLeaderBoards
            Case Is = "ucEditQuestions"
                newUC = New ucEditQuestions 'Loads the user control ucEditQuestions
        End Select
        Me.Controls.Add(newUC) 'Adds the desired User Control
    End Sub

    Public Sub CloseBtn_Click(sender As Object, e As EventArgs) Handles CloseBtn.Click

        'This IF statement asks the user whether they are sure if they want to leave
        If MsgBox("Are you sure you want to exit the program? ", MsgBoxStyle.YesNo,
        "Exit") = MsgBoxResult.Yes Then

```

```

        End 'This button closes the whole program
    End If

    End Sub
End Class

Public Module Settings

    'The module opens the database and creates a datatable in vb, then any edits that
    occur during the program then
    'it updates the database With the datatable
    Public Function runSQL(ByVal query As String) As DataTable
        Dim con As New OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=dbPMQuiz.mdb")
            'Establishes a connection to a access file by the name dbPMQuiz
            Dim da As OleDb.OleDbDataAdapter
            'This is the data adapter, this fetches data from the database and then
            translates between the database and VB
            'But it does not completely exist at this point
            Dim dt As New DataTable
            'Create the data table variable assigned to dt
            con.Open() 'Opens the database connection
            da = New OleDb.OleDbDataAdapter(query, con) 'The data adapter now exists and it
            runs the following query
            'by the connection
            dt.Clear() 'The table is cleared incase it already had values within it
            dt.Columns.Clear() 'Verification that the table is cleared
            da.Fill(dt) 'The data adapter is now filled with the database
            con.Close() 'Closes the connection to the database
            'It is important to close the connection because only one connection can be made
            at one time
            Return dt 'The data table filled with the database is now outputted from the
            function
        End Function
    End Module

Public Class ucAbout
    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

        Dim ucLoad As MainForm

        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
        access AddUC
        ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which
        determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control

    End Sub
End Class

Public Class ucAdminMainMenu

    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

```

Candidate Name: Naveed Ali Rafeeq Candidate Number: 1904

```

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

Private Sub LeadBtn_MouseEnter(sender As Object, e As EventArgs) Handles
Leadbtn.MouseEnter
    Leadbtn.BackgroundImage = My.Resources.BgBlurSmall
    'This sub loads a different image when the mouse hovers over the Maths Quiz
button
End Sub

Private Sub EditBtn_MouseEnter(sender As Object, e As EventArgs) Handles
EditBtn.MouseEnter
    EditBtn.BackgroundImage = My.Resources.BgBlurSmall
    'Same as the above but with the Edit Questions button Quiz button
End Sub

Private Sub Leadbtn_Click(sender As Object, e As EventArgs) Handles Leadbtn.Click

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucLeaderBoards") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

Private Sub EditBtn_Click(sender As Object, e As EventArgs) Handles EditBtn.Click

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucEditQuestions") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub
End Class

Public Class CreateAccBtn
    Private Sub NewAccBtn_Click(sender As Object, e As EventArgs) Handles NewAccBtn.Click
        'All the if statements in this sub routine are forms of validation which stop the
user from creating a new account

        Dim dtCreateAcc As DataTable 'Creates the table dtCreateAcc, a copy of the
student table in the database

        If NPassTxt.Text <> NPass2Txt.Text Then
            MsgBox("Password do not match")
            Exit Sub
        End If
    End Sub

```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

'If the text in the NPass textbox and NPass2 textbox do not match then it exits the sub routine

```
dtCreateAcc = runSQL("Select * from AdminData where TID = '" & TIDTxt.Text & "'")
If dtCreateAcc.Rows.Count <> 1 Then
    MsgBox("Incorrect Teacher ID")
    Exit Sub
End If
'If the text in the TID textbox is not in the DataTable then it exits the sub routine

dtCreateAcc = runSQL("Select * from StudentData where SUserN = '" &
NusrNmeTxt.Text & "'")
If dtCreateAcc.Rows.Count > 0 Then
    MsgBox("Account name already in use, please use another")
    Exit Sub
'If the text in the NusrNme textbox is already in the SUserN column of the
DataTable then it exits the sub routine
Else
    dtCreateAcc = runSQL("Insert into StudentData (SUserN, SPass, TID) values ('"
& NusrNmeTxt.Text &
        "','" & NPassTxt.Text & "','" & TIDTxt.Text & "')")
    MsgBox("New Account Added, you can now login")
End If
'If it is not already in that column then it adds the text from the three text
boxes into the student table in the database

Dim ucLoad As MainForm

ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which
determines which user control to load
Me.Dispose() 'Gets rid of the user control
GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

'Does a check to see if a key has been pressed when the user is in the password
textbox
Private Sub TIDTxt_KeyDown(sender As Object, e As KeyEventArgs) Handles
TIDTxt.KeyDown
    If e.KeyCode = Keys.Enter Then
        Call NewAccBtn_Click(sender, e)
    End If 'When the user presses enter button in the password textbox it runs the
login button

End Sub

Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
    Dim ucLoad As MainForm

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control
```

```

    End Sub

End Class

Public Class ucEditQuestions

    Dim ucLoad As MainForm
    Dim DeleteUser As String 'The Username of the account the admin wishes to delete

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
        ucLoad.addUC("ucAdminMainMenu") 'Puts this string into the subroutine AddUC which
determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control

    End Sub

    Private Sub UserList() 'Sub that loads all students in with matching TID's as the
admin

        Dim dt As New DataTable 'Copies a table within the database into vb to be edited
and then updated

        dt = runSQL("SELECT SUserName, SPass from StudentData where TID = '" &
Details.UserTID & "' ORDER BY SUserName ASC;")
        'Loads all the users with their usernames and passwords into the DataTable

        For x = 0 To dt.Rows.Count - 1
            Dim items(1) As String '1 dimensional array with 2 values
            items(0) = dt.Rows(x)(0) 'Username
            items(1) = dt.Rows(x)(1) 'Password
            'Adds the username and password into the row

            Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items
array into the LineNew
            UserLst.Items.Add(LineNew) 'Adds the new line into the List View
        Next 'Loop repeats for each row of the table
    End Sub

    Private Sub MQuestionsList() 'Sub that loads all the questions and answers from the
MQuestions table into the List View

        Dim dt As New DataTable 'Copies a table within the database into vb to be edited
and then updated
        dt = runSQL("SELECT * FROM MQuestions ORDER BY MQuestion ASC;")
        'Loads everything from the MQuestions table in ascending order

        For x = 0 To dt.Rows.Count - 1
            Dim items(4) As String '1 dimensional array with 5 values
            items(0) = dt.Rows(x)(0) 'Question
            items(1) = dt.Rows(x)(1) 'Correct Answer
            items(2) = dt.Rows(x)(2) 'Answer 2
            items(3) = dt.Rows(x)(3) 'Answer 3
            items(4) = dt.Rows(x)(4) 'Answer 4

```

```

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items
array into the LineNew
        MQtnLst.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub

Private Sub PQuestionsList() 'Sub that loads all the questions and answers from the
PQuestions table into the List View

    Dim dt As New DataTable 'Copies a table within the database into vb to be edited
and then updated
    dt = runSQL("SELECT * FROM PQuestions ORDER BY PQuestion ASC;")
    'Loads everything from the PQuestions table in ascending order

    For x = 0 To dt.Rows.Count - 1
        Dim items(4) As String '1 dimensional array with 5 values
        items(0) = dt.Rows(x)(0) 'Question
        items(1) = dt.Rows(x)(1) 'Correct Answer
        items(2) = dt.Rows(x)(2) 'Answer 2
        items(3) = dt.Rows(x)(3) 'Answer 3
        items(4) = dt.Rows(x)(4) 'Answer 4

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items
array into the LineNew
        PQtnLst.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
DeleteUserBtn.Click
    'Button that deletes everything of the user selected

    runSQL("DELETE * FROM StudentData WHERE SUserName = '" & DeleteUser & "'") 'Deletes
whole row from StudentData table
    runSQL("DELETE * FROM MResults WHERE SUserName = '" & DeleteUser & "'") 'Deletes
whole row from MResults table
    runSQL("DELETE * FROM PResults WHERE SUserName = '" & DeleteUser & "'") 'Deletes
whole row from PResult table
    MsgBox("User has been deleted")

    'Updates the List View
    UserLst.Items.Clear() 'Clear all the items from the List View
    UserList() 'Loads all the items again
End Sub

Private Sub MRefreshBtn_Click(sender As Object, e As EventArgs) Handles
MRefreshBtn.Click
    'This sub updates the Maths question by deleting it from the Maths Question table
then adding it again

    Try
        runSQL("DELETE * FROM MQuestions WHERE MQuestion = '" &
MQtnLst.SelectedItems(0).SubItems(0).Text & "'")
        runSQL("INSERT into MQuestions (MQuestion, MA1, MA2, MA3, MA4) values ('" &
Qntxt.Text & "','" &
A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" &
A4txt.Text & "')")
    
```

```

Candidate Name: Naveed Ali Rafeeq Candidate Number: 1904
    'Deletes the whole row in the Maths Question table that contains the question
selected
    'Adds the question and the 4 answers within the textboxes into the same table

    'Updates the listview
    MQtnLst.Items.Clear() 'Deletes all the items from the listview
    MQuestionsList() 'Runs the sub routine to load the new MQuestion table
    MsgBox("Question Edited!") 'Message box informing the user that the question
has been updated

    Catch ex As Exception
        'The try statement allows the program not to crash when an item in the
listview is not selected
        'It works by the program trying to perform the the sql instructions but if it
can't then it just makes the button do nothing
    End Try
End Sub

Private Sub PRefreshBtn_Click(sender As Object, e As EventArgs) Handles
PRefreshBtn.Click
    'This sub updates the Physics question by deleting it from the Physics Question
table then adding it again

    Try
        runSQL("DELETE * FROM PQuestions WHERE PQuestion = '" &
PQtnLst.SelectedItems(0).SubItems(0).Text & "'")
        runSQL("INSERT into PQuestions (PQuestion, PA1, PA2, PA3, PA4) values ('" &
Qtntxt.Text & "','" &
A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" &
A4txt.Text & "')")
        'Deletes the whole row in the Physics Question table that contains the
question selected
        'Adds the question and the 4 answers within the textboxes into the same table

        'Updates the listview
        PQtnLst.Items.Clear() 'Deletes all the items from the listview
        PQuestionsList() 'Runs the sub routine to load the new PQuestion table
        MsgBox("Question Edited!") 'Message box informing the user that the question
has been edited

        Catch ex As Exception
            'The try statement allows the program not to crash when an item in the
listview is not selected
            'It works by the program trying to perform the the sql instructions but if it
can't then it just makes the button do nothing
        End Try
    End Sub

Private Sub MQtnAddBtn_Click(sender As Object, e As EventArgs) Handles
MQtnAddBtn.Click
    'Sub that adds the Maths Question to the Maths Questions table when they click
the Add button

    Try
        runSQL("INSERT into MQuestions (MQuestion, MA1, MA2, MA3, MA4) values ('" &
Qtntxt.Text & "','" &
A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" &
A4txt.Text & "')")

```

'Adds the question and the 4 answers within the textboxes into the Maths Question table

```

'Updates the listview
MQtnLst.Items.Clear() 'Deletes all the items from the listview
MQuestionsList() 'Runs the sub routine to load the new MQuestion table
MsgBox("Question Added!") 'Message box informing the user that the question
has been added

Catch ex As Exception
    'The try statement allows the program not to crash when an item in the
listview is not selected
    'It works by the program trying to perform the the sql instructions but if it
can't then it just makes the button do nothing
End Try
End Sub

Private Sub PQtnAddBtn_Click(sender As Object, e As EventArgs) Handles
PQtnAddBtn.Click
    'Sub does the same as the one above but with the Physics questions table

    Try
        runSQL("INSERT into PQuestions (PQuestion, PA1, PA2, PA3, PA4) values ('" &
Qtntxt.Text & "','" &
A1txt.Text & "','" & A2txt.Text & "','" & A3txt.Text & "','" & A4txt.Text &
"')")
        PQtnLst.Items.Clear()
        PQuestionsList()
        MsgBox("Question Added!")

    Catch ex As Exception
    End Try
End Sub

Private Sub MQtnDeleteBtn_Click(sender As Object, e As EventArgs) Handles
MQtnDeleteBtn.Click
    'Sub that deletes the Maths question from the Maths Question Table

    Dim DeletedCheck As Boolean = False 'Creates the false boolean value that will be
used to with deleting the user

    Try
        If MsgBox("Are you sure you want to delete this question? ",_
MsgBoxStyle.YesNo, "Delete Question") = MsgBoxResult.Yes Then
            runSQL("DELETE * FROM MQuestions WHERE MQuestion = '" &
MQtnLst.SelectedItems(0).SubItems(0).Text & "'")
            'If the user chooses the yes option in the message box then
            'It deletes the whole row from the Maths Question table that contains the
question selected in the list view
            DeletedCheck = True 'Turns the boolean variable into true

            'Updates the listview
            MQtnLst.Items.Clear() 'Deletes all the items from the listview
            MQuestionsList() 'Runs the sub routine to load the new MQuestion table
        End If
        'If they choose No then the question would not be deleted and the message box
will close
    End Try
End Sub

```

```

        If DeletedCheck = True Then
            MsgBox("Question Deleted!")
        End If
        'If the user choose yes in the user box it shows this message box afterwards
        informing the user that the question has been deleted

        Catch ex As Exception
            'The try statement allows the program not to crash when an item in the
            listview is not selected
            'It works by the program trying to perform the the sql instructions but if it
            can't then it just makes the button do nothing
        End Try
    End Sub

    Private Sub PQtnDeleteBtn_Click(sender As Object, e As EventArgs) Handles
PQtnDeleteBtn.Click
        'Same sub as above but deletes from the Physics Question table
        Dim DeletedCheck As Boolean = False

        Try
            If MsgBox("Are you sure you want to delete this question? ",
MsgBoxStyle.YesNo, "Delete Question") = MsgBoxResult.Yes Then
                runSQL("DELETE * FROM PQuestions WHERE PQuestion = '" &
PQtnLst.SelectedItems(0).SubItems(0).Text & "'")
                DeletedCheck = True
                PQtnLst.Items.Clear()
                PQuestionsList()
            End If

            If DeletedCheck = True Then
                MsgBox("Question Deleted!")
            End If

            Catch ex As Exception
        End Try
    End Sub

    Private Sub MQtnLst_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
MQtnLst.Click
        'Sub that fills the textboxes with the row clicked in the Maths List View

        Qtntxt.Text = MQtnLst.SelectedItems(0).SubItems(0).Text 'Fills textbox with the
        Question
        A1txt.Text = MQtnLst.SelectedItems(0).SubItems(1).Text 'Fills textbox with the
        Right Answer
        A2txt.Text = MQtnLst.SelectedItems(0).SubItems(2).Text 'Fills textbox with Answer
2
        A3txt.Text = MQtnLst.SelectedItems(0).SubItems(3).Text 'Fills textbox with Answer
3
        A4txt.Text = MQtnLst.SelectedItems(0).SubItems(4).Text 'Fills textbox with Answer
4

        End Sub

    Private Sub PQtnLst_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
PQtnLst.Click
        'Sub that fills the textboxes with the row clicked in the Physics List View

```

```

        Qtntxt.Text = PQtnLst.SelectedItems(0).SubItems(0).Text 'Fills textbox with the
Question
        A1txt.Text = PQtnLst.SelectedItems(0).SubItems(1).Text 'Fills textbox with the
Right Answer
        A2txt.Text = PQtnLst.SelectedItems(0).SubItems(2).Text 'Fills textbox with Answer
2
        A3txt.Text = PQtnLst.SelectedItems(0).SubItems(3).Text 'Fills textbox with Answer
3
        A4txt.Text = PQtnLst.SelectedItems(0).SubItems(4).Text 'Fills textbox with Answer
4

    End Sub

    Private Sub UsrLst_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
UserLst.Click
        DeleteUser = UserLst.SelectedItems(0).SubItems(0).Text 'The selected users
username is now equal to the variable
    End Sub

    Private Sub UsrLstBtn_MouseEnter(sender As Object, e As EventArgs) Handles
UsrLstBtn.MouseEnter
        UsrLstBtn.BackgroundImage = My.Resources.BgBlurSmall
        'This sub loads a different image when the mouse hovers over the Delete User
button
    End Sub

    Private Sub MQtnBtn_MouseEnter(sender As Object, e As EventArgs) Handles
MQtnbtn.MouseEnter
        MQtnbtn.BackgroundImage = My.Resources.BgBlurSmall
        'Same as the above but with the Edit Maths Questions button
    End Sub

    Private Sub PQtnBtn_MouseEnter(sender As Object, e As EventArgs) Handles
PQtnbtn.MouseEnter
        PQtnbtn.BackgroundImage = My.Resources.BgBlurSmall
        'Same as the above but with the Edit Physics Questions button
    End Sub

    Private Sub ucEditQuestions_Load(sender As Object, e As EventArgs) Handles
 MyBase.Load
        'When the UC loads it hides the List Views and the Back to Edit Page button

        UsrLstHide() 'Sub that hides the User List View and its delete button
        MQtnHide() 'Sub that hides the Maths Question List View, the text boxes and its
buttons
        PQtnHide() 'Sub that hides the Physics Question List View, the text boxes and its
buttons
        EditBtn.Visible = False 'Edit button is hidden
    End Sub

    Private Sub EditBtn_Click(sender As Object, e As EventArgs) Handles EditBtn.Click
        'Sub that hides all the list views and makes the menu buttons visible when the
Edit button is clicked

        BtnsShow() 'Sub that makes the menu buttons visible
        UsrLstHide() 'Sub that hides the User List View and its delete button

```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

```
MQtnHide() 'Sub that hides the Maths Question List View, the text boxes and its
buttons
PQtnHide() 'Sub that hides the Physics Question List View, the text boxes and its
buttons
    BackBtn.Visible = True 'Back button is visible
    EditBtn.Visible = False 'Edit button is hidden
    TitleLbl.Location = New Point(263, 27) 'Moves the Title back to its original
position
End Sub

Private Sub UsrLstBtn_Click(sender As Object, e As EventArgs) Handles UsrLstBtn.Click
    'Sub that loads the User List when the Delete User button is clicked

    UsrLstShow() 'Sub that loads the User List View and its delete button
    BtnsHide() 'Sub that hides the menu buttons
    EditBtn.Visible = True 'Edit button is visible
    BackBtn.Visible = False 'Back button is hidden
End Sub

Private Sub MQtnbtn_Click(sender As Object, e As EventArgs) Handles MQtnbtn.Click
    'Sub that loads the Maths Question List View, the text boxes and its buttons when
the
    'Edit Maths Questions button is clicked

    MQtnShow() 'Sub that loads the Maths User List View and its delete button
    BtnsHide() 'Sub that hides the menu buttons
    EditBtn.Visible = True 'Edit button is visible
    BackBtn.Visible = False 'Back button is hidden
    TitleLbl.Location = New Point(15, 35) 'Moves the title to a new position
End Sub

Private Sub PQtnbtn_Click(sender As Object, e As EventArgs) Handles PQtnbtn.Click
    'Sub that does the same as above but with the Physics counterpart
    PQtnShow()
    BtnsHide()
    EditBtn.Visible = True
    BackBtn.Visible = False
    TitleLbl.Location = New Point(15, 35)
End Sub

Private Sub BtnsShow() 'Sub that loads the menu buttons
    UsrLstBtn.Visible = True
    MQtnbtn.Visible = True
    PQtnbtn.Visible = True
End Sub

Private Sub BtnsHide() 'Sub that hides the menu buttons
    UsrLstBtn.Visible = False
    MQtnbtn.Visible = False
    PQtnbtn.Visible = False
End Sub

Private Sub UsrLstHide() 'Sub that hides the User List and its delete user button
    UserLst.Visible = False
    DeleteUserBtn.Visible = False
    UserLst.Items.Clear()
End Sub
```

```

Private Sub UsrLstShow() 'Sub that loads the User List and its delete user button
    UserLst.Visible = True
    DeleteUserBtn.Visible = True
    UserList()
End Sub

Private Sub MQtnHide() 'Sub that hides the Maths Question List View, the text boxes
and its buttons
    MQtnLst.Visible = False
    Qtntxt.Visible = False
    A1txt.Visible = False
    A2txt.Visible = False
    A3txt.Visible = False
    A4txt.Visible = False
    QtnLbl.Visible = False
    A1Lbl.Visible = False
    A2Lbl.Visible = False
    A3Lbl.Visible = False
    A4Lbl.Visible = False
    MRefreshBtn.Visible = False
    MQtnAddBtn.Visible = False
    MQtnDeleteBtn.Visible = False
    MQtnLst.Items.Clear() 'Clears all the items from the List View so that duplicate
values will not
        'be there when it is next loaded
End Sub

Private Sub MQtnShow() 'Sub that loads the Maths Question List View, the text boxes
and its buttons
    MQtnLst.Visible = True
    Qtntxt.Visible = True
    A1txt.Visible = True
    A2txt.Visible = True
    A3txt.Visible = True
    A4txt.Visible = True
    QtnLbl.Visible = True
    A1Lbl.Visible = True
    A2Lbl.Visible = True
    A3Lbl.Visible = True
    A4Lbl.Visible = True
    MRefreshBtn.Visible = True
    MQtnAddBtn.Visible = True
    MQtnDeleteBtn.Visible = True
    MQuestionsList() 'Sub that fills the List View with the Maths Questions
End Sub

Private Sub PQtnHide() 'Sub that hides the Physics Question List View, the text boxes
and its buttons
    PQtnLst.Visible = False
    Qtntxt.Visible = False
    A1txt.Visible = False
    A2txt.Visible = False
    A3txt.Visible = False
    A4txt.Visible = False
    QtnLbl.Visible = False
    A1Lbl.Visible = False
    A2Lbl.Visible = False

```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

```
A3Lbl.Visible = False
A4Lbl.Visible = False
PRefreshBtn.Visible = False
PQtnAddBtn.Visible = False
PQtnDeleteBtn.Visible = False
PQtnLst.Items.Clear() 'Clears all the items from the List View so that duplicate
values will not
    'be there when it is next loaded
End Sub

Private Sub PQtnShow() 'Sub that loads the Physics Question List View, the text boxes
and its buttons
    PQtnLst.Visible = True
    Qtntxt.Visible = True
    A1txt.Visible = True
    A2txt.Visible = True
    A3txt.Visible = True
    A4txt.Visible = True
    QtnLbl.Visible = True
    A1Lbl.Visible = True
    A2Lbl.Visible = True
    A3Lbl.Visible = True
    A4Lbl.Visible = True
    PRefreshBtn.Visible = True
    PQtnAddBtn.Visible = True
    PQtnDeleteBtn.Visible = True
    PQuestionsList() 'Sub that fills the List View with the Physics Questions
End Sub
End Class

Public Class ucLeaderBoards

    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
        ucLoad.addUC("ucAdminMainMenu") 'Puts this string into the subroutine AddUC which
determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    End Sub

    Private Sub MathsScoreLoad() 'Sub that places all of the Usernames and their adjacent
Maths scores into the List View

        Dim dt As New DataTable 'Copies a table within the database into vb to be edited
and then updated
        dt = runSQL("SELECT * from MResults where TID = '" & Details.UserTID & "' ORDER
BY Results DESC;")
            'Loads all the rows that have the same TID as the one used to log in and places
them into the Data Table

        For x = 0 To dt.Rows.Count - 1
            Dim items(1) As String '1 dimensional array with 2 values
            items(0) = dt.Rows(x)(0) 'Username
```

```

        items(1) = dt.Rows(x)(1) 'Score
        'Adds the username and password into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items
array into the LineNew
        MLstScore.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub

Private Sub PhysicsScoreLoad() 'Sub that places all of the Usernames and their
adjacent Physics scores into the List View

    Dim dt As New DataTable
    dt = runSQL("SELECT * from PResults where TID = '" & Details.UserTID & "' ORDER
BY Results DESC;")
        'Loads all the rows that have the same TID as the one used to log in and places
them into the Data Table

    For x = 0 To dt.Rows.Count - 1
        Dim items(1) As String '1 dimensional array with 2 values
        items(0) = dt.Rows(x)(0) 'Username
        items(1) = dt.Rows(x)(1) 'Score
        'Adds the username and password into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items
array into the LineNew
        PLstScore.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table
End Sub

Private Sub ucLeaderBoards_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    'When the UC loads it runs the two subs that places the items within the
leaderboards

    MathsScoreLoad()
    PhysicsScoreLoad()

End Sub
End Class

Public Class ucLogin

    Dim ucLoad As MainForm

    'Does a check to see if a key has been pressed when the user is in the password
textbox
    Private Sub PassTxt_PreviewKeyDown(sender As Object, e As PreviewKeyDownEventArgs)
Handles PassTxt.PreviewKeyDown
        If e.KeyCode = Keys.Enter Then
            Call LoginBtn_Click(sender, e)
        End If 'When the user presses enter button in the password textbox it runs the
login button
    End Sub

    Private Sub LoginBtn_Click(sender As Object, e As EventArgs) Handles LoginBtn.Click

        Dim dtAdmin, dtStudent As New DataTable 'Creates 2 datatable variables

```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

```
        dtAdmin = runSQL("Select * from AdminData where TUserN = '" & UsrNmeTxt.Text & "'  
and TPass = '" & PassTxt.Text & "')  
        dtStudent = runSQL("Select * from StudentData where SUserN = '" & UsrNmeTxt.Text  
& "' and SPass = '" & PassTxt.Text & "')  
        'Extracts all the Usernames and Passwords from the Admin and Student Tables  
  
        If dtStudent.Rows.Count = 1 Then  
            ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows  
ucLoad to access AddUC  
            ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which  
determines which user control to load  
            Me.Dispose() 'Gets rid of the user control  
            GC.Collect() 'Gather all the remaining data left after closing the user  
control  
            Details.LoginUser = dtStudent.Rows(0)(0) 'Saves the SUserN on the student row  
to be used later  
            Details.UserTID = dtStudent.Rows(0)(2) 'Saves the TID on the student row to  
be used later  
        ElseIf dtAdmin.Rows.Count = 1 Then  
            ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows  
ucLoad to access AddUC  
            ucLoad.addUC("ucAdminMainMenu") 'Puts this string into the subroutine AddUC  
which determines which user control to load  
            Me.Dispose() 'Gets rid of the user control  
            GC.Collect() 'Gather all the remaining data left after closing the user  
control  
            Details.LoginUser = dtAdmin.Rows(0)(1) 'Saves the TUserN on the admin row to  
be used later  
            Details.UserTID = dtAdmin.Rows(0)(0) 'Saves the TID on the admin row to be  
used later  
        Else  
            MsgBox("Your details have been entered incorrectly, please try again")  
        End If  
  
    End Sub  
  
    Private Sub NewAccBtn_Click(sender As Object, e As EventArgs) Handles NewAccBtn.Click  
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to  
access AddUC  
        ucLoad.addUC("ucCreateAcc") 'Puts this string into the subroutine AddUC which  
determines which user control to load  
        Me.Dispose() 'Gets rid of the user control  
        GC.Collect() 'Gather all the remaining data left after closing the user control  
    End Sub  
End Class  
  
Public Class ucMainMenu  
  
    Dim ucLoad As MainForm  
  
    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click  
  
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to  
access AddUC  
        ucLoad.addUC("ucLogin") 'Puts this string into the subroutine AddUC which  
determines which user control to load  
        Me.Dispose() 'Gets rid of the user control  
        GC.Collect() 'Gather all the remaining data left after closing the user control  
    End Sub
```

```

    End Sub

    Private Sub MQuizbtn_MouseEnter(sender As Object, e As EventArgs) Handles MQuizbtn.MouseEnter
        MQuizbtn.BackgroundImage = My.Resources.BgBlurSmall
        'This sub loads a different image when the mouse hovers over the Maths Quiz
button
    End Sub

    Private Sub PQuizbtn_MouseEnter(sender As Object, e As EventArgs) Handles PQuizbtn.MouseEnter
        PQuizbtn.BackgroundImage = My.Resources.BgBlurSmall
        'Same as the above but with the Physics Quiz button
    End Sub

    Private Sub Aboutbtn_MouseEnter(sender As Object, e As EventArgs) Handles Aboutbtn.MouseEnter
        Aboutbtn.BackgroundImage = My.Resources.BgBlurSmall
        'Same as the above two but with the About button
    End Sub

    Private Sub MQuizbtn_Click(sender As Object, e As EventArgs) Handles MQuizbtn.Click

        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
        ucLoad.addUC("ucMathsQuiz") 'Puts this string into the subroutine AddUC which
determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control

    End Sub

    Private Sub Aboutbtn_Click(sender As Object, e As EventArgs) Handles Aboutbtn.Click

        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
        ucLoad.addUC("ucAbout") 'Puts this string into the subroutine AddUC which
determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control

    End Sub

    Private Sub ucMainMenu_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Details.QtnTotal = 0 'This resets the variable question total back to 0
        Details.QtnRight = 0 'This resets the variable question right back to 0
    End Sub

    Private Sub PQuizbtn_Click(sender As Object, e As EventArgs) Handles PQuizbtn.Click

        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
        ucLoad.addUC("ucPhysicsQuiz") 'Puts this string into the subroutine AddUC which
determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user control
    End Sub

```

```

    End Sub
End Class

```

```
Public Class ucMathsFeedback
```

```
    Dim ucLoad As MainForm
```

```
    Private Sub ucMathsFeedback_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        QtnRLbl.Text = Details.QtnRight 'The total questions the user got correct is assigned to the label
```

```
        Select Case Details.QtnRight 'The select case assigns a grade and a message to each value of QtnRight global variable
```

```
            Case Is > 17
```

```
                GrdLbl.Text = "A*"
```

```
                FBackLbl.Text = "You're A Genius"
```

```
            Case Is = 16 Or 17
```

```
                GrdLbl.Text = "A"
```

```
                FBackLbl.Text = "Pretty Good"
```

```
            Case Is = 14 Or 15
```

```
                GrdLbl.Text = "B"
```

```
                FBackLbl.Text = "Decent"
```

```
            Case Is = 12 Or 13
```

```
                GrdLbl.Text = "C"
```

```
                FBackLbl.Text = "You Passed"
```

```
            Case Is = 10 Or 11
```

```
                GrdLbl.Text = "D"
```

```
                FBackLbl.Text = "You Can Do Better"
```

```
            Case Is = 8 Or 9
```

```
                GrdLbl.Text = "E"
```

```
                FBackLbl.Text = "Unlucky"
```

```
            Case Is < 8
```

```
                GrdLbl.Text = "U"
```

```
                FBackLbl.Text = "Do the test again"
```

```
        End Select
```

```
        LstScore.Visible = False 'The list view is hidden
```

```
        FeedBtn.Visible = False 'The button to send you back to the feedback is hidden
```

```
        Dim dtScore As New DataTable 'Copies a table within the database into vb to be edited and then updated
```

```
        dtScore = runSQL("SELECT * from MResults where SUserN = '" & Details.LoginUser & "'")
```

```
        'Loads the query, which is the row equal to the username used when logging in, into the Data Table
```

```
        If dtScore.Rows.Count > 0 Then 'If there is atleast a row in the datatable
```

```
            runSQL("DELETE * FROM MResults where SUserN = '" & Details.LoginUser & "'")
```

```
            'Delete the row
```

```
            runSQL("INSERT into MResults (SUserN, Results, TID) values ('" & Details.LoginUser & "','" &
```

```
                Details.QtnRight & "','" & Details.UserTID & "')")
```

```
            'Add the new row with the same username, TID but the updated score
```

```
        Else 'If there is no rows then
```

```
            runSQL("INSERT into MResults (SUserN, Results, TID) values ('" & Details.LoginUser & "','" &
```

```
                Details.QtnRight & "','" & Details.UserTID & "')")
```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

```
        Details.QtnRight & "','" & Details.UserTID & "')")
        'Add the new row with the username, TID and score
    End If

    ScoreLoad() 'Places the items into the List View

End Sub

Private Sub ScoreLoad() 'Sub that loads the items (Username and Physics score) into
the List View

    Dim dt As New DataTable 'Copies a table within the database into vb to be edited
and then updated
    dt = runSQL("SELECT * from MResults where TID = '" & Details.UserTID & "' ORDER
BY Results DESC;")
        'Loads all the rows from the Physics Results table that are in a certain class
into the Data Table
        'Orders them from highest to lowest, therefore making it a leader board

    For x = 0 To dt.Rows.Count - 1
        Dim items(1) As String '1 dimensional array with 2 values
        items(0) = dt.Rows(x)(0) 'Username
        items(1) = dt.Rows(x)(1) 'Score
        'Adds the username and score into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items
array into the LineNew
        LstScore.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table

End Sub

Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

Private Sub LeadBtn_Click(sender As Object, e As EventArgs) Handles LeadBtn.Click

    'When the Leader Boards button is clicked the sub hides all the feeback text and
the Leader board
    'button, then makes the listview and the Feedback button visible
    LstScore.Visible = True
    FBackLbl.Visible = False
    Label2.Visible = False
    GrdLbl.Visible = False
    QtnRLbl.Visible = False
    Label3.Visible = False
    FeedBtn.Visible = True
    LeadBtn.Visible = False

End Sub
```

```

    Private Sub FeedBtn_Click(sender As Object, e As EventArgs) Handles FeedBtn.Click
        'When the Leader Boards button is clicked the sub hides the listview and the
        'Feedback button,
        'then makes the Feedback text and the Leader Board button visible
        LstScore.Visible = False
        FBackLbl.Visible = True
        Label2.Visible = True
        GrdLbl.Visible = True
        QtnRlbl.Visible = True
        Label3.Visible = True
        FeedBtn.Visible = False
        LeadBtn.Visible = True

    End Sub
End Class

Public Class ucMathsQuiz

    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click
        If MsgBox("Are you sure you want to leave, your quiz score won't be saved? ", _
        MsgBoxStyle.YesNo, "Maths Quiz") = MsgBoxResult.Yes Then
            ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows
            ucLoad to access AddUC
            ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which
            determines which user control to load
            Me.Dispose() 'Gets rid of the user control
            GC.Collect() 'Gather all the remaining data left after closing the user
            control
        End If
    End Sub

    Private Sub ucMathsQuiz_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim dtLdQtn As DataTable 'Assigns the datatable to a variable
        Dim RowAmount, ranNum As Integer 'Creates 2 integer variables

        dtLdQtn = runSQL("Select * from MQuestions") 'Loads the MQuestion field from the
        MQuestions table in the access database
        RowAmount = dtLdQtn.Rows.Count 'Counts the amount of rows in the table and
        assigns the integer to RowAmount
        Randomize()
        ranNum = Rnd() * (RowAmount - 1) 'This generates a random number from how many
        rows there was
        QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random
        question from the table

        Dim ARanArray() As Integer = {1, 2, 3, 4} 'Creates an Array full of integers
        between 1 and 4
        Dim tempAns As String 'Creates the temporary string variable
        Dim ARnd1, ARnd2 As Integer 'Creates 2 integer variables

```

Candidate Name: Naveed Ali Rafeeq  
For x = 0 To 9 'Loops 10 times

Candidate Number: 1904

```
    Randomize()
    ARnd1 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assigns
it to the first integer variable
    Randomize()
    ARnd2 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assangs
it to the second integer variable

        tempAns = ARanArray(ARnd1) 'The string variable now contains the a part of
the array thats a value of rnd1 along
        ARanArray(ARnd1) = ARanArray(ARnd2) 'The value that was contained in that
part of the array is now equal to the value that was in the array rnd2 along
        ARanArray(ARnd2) = tempAns 'Now the value of the original rn1 is now placed
into the array rnd2 along
            'It has now swapped the two numbers around within the array
        Next
        'This loop therefore swaps different numbers 10 times essentially a shuffling the
array

        Rbtn1.Text = dtLdQtn.Rows(ranNum)(ARanArray(0))
        RBtn2.Text = dtLdQtn.Rows(ranNum)(ARanArray(1))
        RBtn3.Text = dtLdQtn.Rows(ranNum)(ARanArray(2))
        RBtn4.Text = dtLdQtn.Rows(ranNum)(ARanArray(3))
        'Loads all the answers within the array and equals them to the radio button
text's
        Details.QtnTotal += 1
        QNumLbl.Text = CStr(Details.QtnTotal)
        FeedBackLd()

    End Sub

Public Sub AnsBtn_Click(sender As Object, e As EventArgs) Handles AnsBtn.Click

    Dim dtLdAns As New DataTable

    If Rbtn1.Checked Then 'If the 1st radio button is ticked then
        dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & Rbtn1.Text & "'")
    'Loads from the database the rows which have the MA1 value equal to the radiobutton text
        If dtLdAns.Rows.Count > 0 Then 'If there is more than 0 rows left then it
outputs a message saying they are correct
            MsgBox("                               CORRECT!
                                         Onto The Next Question   ")
            Details.QtnRight += 1 'If this message box is displayed then the question
right variable increments
        Else
            MsgBox("                               INCORRECT!
                                         Onto The Next Question   ")
        End If
    End If

    If RBtn2.Checked Then 'Same but with the 2nd radio button
        dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & RBtn2.Text & "'")
        If dtLdAns.Rows.Count > 0 Then
            MsgBox("                               CORRECT!
```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

```
        Onto The Next Question    ")
        Details.QtnRight += 1
    Else
        MsgBox("                                INCORRECT!

            Onto The Next Question    ")
        End If
    End If

If RBtn3.Checked Then 'Same but with the 3rd radio button
    dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & RBtn3.Text & "'")
    If dtLdAns.Rows.Count > 0 Then
        MsgBox("                                CORRECT!

            Onto The Next Question    ")
        Details.QtnRight += 1
    Else
        MsgBox("                                INCORRECT!

            Onto The Next Question    ")
        End If
    End If

If RBtn4.Checked Then 'Same but with the 4th radio button
    dtLdAns = runSQL("Select * from MQuestions where MA1 = '" & RBtn4.Text & "'")
    If dtLdAns.Rows.Count > 0 Then
        MsgBox("                                CORRECT!

            Onto The Next Question    ")
        Details.QtnRight += 1
    Else
        MsgBox("                                INCORRECT!

            Onto The Next Question    ")
        End If
    End If

ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucMathsQuiz") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

Private Sub FeedBackLd()
    If Details.QtnTotal = 21 Then
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows
ucLoad to access AddUC
        ucLoad.addUC("ucMathsFeedback") 'Puts this string into the subroutine AddUC
which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user
control
    End If
End Sub
```

```
Public Class ucPhysicsFeedback

    Dim ucLoad As MainForm

    Private Sub ucPhysicsFeedback_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        QtnRLbl.Text = Details.QtnRight 'The total questions the user got correct is assigned to the label

        Select Case Details.QtnRight 'The select case assigns a grade and a message to each value of QtnRight global variable
            Case Is > 17
                GrdLbl.Text = "A*"
                FBackLbl.Text = "You're A Genius"
            Case Is = 16 Or 17
                GrdLbl.Text = "A"
                FBackLbl.Text = "Pretty Good"
            Case Is = 14 Or 15
                GrdLbl.Text = "B"
                FBackLbl.Text = "Decent"
            Case Is = 12 Or 13
                GrdLbl.Text = "C"
                FBackLbl.Text = "You Passed"
            Case Is = 10 Or 11
                GrdLbl.Text = "D"
                FBackLbl.Text = "Unlucky"
            Case Is = 8 Or 9
                GrdLbl.Text = "E"
                FBackLbl.Text = "Peak"
            Case Is < 8
                GrdLbl.Text = "U"
                FBackLbl.Text = "Do The Test Again"
        End Select

        LstScore.Visible = False 'The list view is hidden
        FeedBtn.Visible = False 'The button to send you back to the feedback is hidden

        Dim dtScore As New DataTable 'Copies a table within the database into vb to be edited and then updated

        dtScore = runSQL("SELECT * from PResults where SUserN = '" & Details.LoginUser & "'")
        'Loads the query, which is the row equal to the username used when logging in, into the Data Table
        If dtScore.Rows.Count > 0 Then 'If there is atleeast a row in the datatable
            runSQL("DELETE * FROM PResults where SUserN = '" & Details.LoginUser & "'")
            'Delete the row
            runSQL("INSERT into PResults (SUserN, Results, TID) values ('" & Details.LoginUser & "','" &
Details.QtnRight & "','" & Details.UserTID & "')")
            'Add the new row with the same username, TID but the updated score
        Else 'If there is no rows then
            runSQL("INSERT into PResults (SUserN, Results, TID) values ('" & Details.LoginUser & "','" &
Details.QtnRight & "','" & Details.UserTID & "')")
        End If
    End Sub
End Class
```

```

        'Add the new row with the username, TID and score
    End If

    ScoreLoad() 'Places the items into the List View

End Sub

Private Sub ScoreLoad() 'Sub that loads the items (Username and Physics score) into
the List View

    Dim dt As New DataTable 'Copies a table within the database into vb to be edited
and then updated
    dt = runSQL("SELECT * from PResults where TID = '" & Details.UserTID & "' ORDER
BY Results DESC;")
        'Loads all the rows from the Physics Results table that are in a certain class
into the Data Table
        'Orders them from highest to lowest, therefore making it a leader board

    For x = 0 To dt.Rows.Count - 1
        Dim items(1) As String '1 dimensional array with 2 values
        items(0) = dt.Rows(x)(0) 'Username
        items(1) = dt.Rows(x)(1) 'Score
        'Adds the username and score into the row

        Dim LineNew As New ListViewItem(items) 'Assigns the contents of the items
array into the LineNew
        LstScore.Items.Add(LineNew) 'Adds the new line into the List View
    Next 'Loop repeats for each row of the table

End Sub

Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

    ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

Private Sub LeadBtn_Click(sender As Object, e As EventArgs) Handles LeadBtn.Click

    'When the Leader Boards button is clicked the sub hides all the feedback text and
the Leader board
    'button, then makes the listview and the Feedback button visible
    LstScore.Visible = True
    FBackLbl.Visible = False
    Label2.Visible = False
    GrdLbl.Visible = False
    QtnRlbl.Visible = False
    Label3.Visible = False
    FeedBtn.Visible = True
    LeadBtn.Visible = False

End Sub

```

Candidate Name: Naveed Ali Rafeeq

Candidate Number: 1904

```
Private Sub FeedBtn_Click(sender As Object, e As EventArgs) Handles FeedBtn.Click

    'When the Leader Boards button is clicked the sub hides the listview and the
    Feedback button,
    'then makes the Feedback text and the Leader Board button visible
    LstScore.Visible = False
    FBackLbl.Visible = True
    Label2.Visible = True
    GrdLbl.Visible = True
    QtnRLbl.Visible = True
    Label3.Visible = True
    FeedBtn.Visible = False
    LeadBtn.Visible = True

End Sub
End Class

Public Class ucPhysicsQuiz

    Dim ucLoad As MainForm

    Private Sub BackBtn_Click(sender As Object, e As EventArgs) Handles BackBtn.Click

        If MsgBox("Are you sure you want to leave, your quiz score won't be saved? ", _
        MsgBoxStyle.YesNo, "Physics Quiz") = MsgBoxResult.Yes Then

            ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows
            ucLoad to access AddUC
            ucLoad.addUC("ucMainMenu") 'Puts this string into the subroutine AddUC which
            determines which user control to load
            Me.Dispose() 'Gets rid of the user control
            GC.Collect() 'Gather all the remaining data left after closing the user
            control

        End If
    End Sub

    Private Sub ucPhysicsQuiz_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Dim dtLdQtn As DataTable 'Assigns the datatable to a variable
        Dim RowAmount, ranNum As Integer 'Creates 2 integer variables

        dtLdQtn = runSQL("Select * from PQuestions") 'Loads the PQuestion field from the
        PQuestions table in the access database
        RowAmount = dtLdQtn.Rows.Count 'Counts the amount of rows in the table and
        assigns the integer to RowAmount
        Randomize()
        ranNum = Rnd() * (RowAmount - 1) 'This generates a random number from how many
        rows there was
        QtnLbl.Text = dtLdQtn.Rows(ranNum)(0) 'The question label prints off a random
        question from the table

        Dim ARanArray() As Integer = {1, 2, 3, 4} 'Creates an Array full of integers
        between 1 and 4
        Dim tempAns As String 'Creates the temporary string variable
        Dim ARnd1, ARnd2 As Integer 'Creates 2 integer variables

        For x = 0 To 9 'Loops 10 times
```

```

        Randomize()
        ARnd1 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assigns
it to the first integer variable
        Randomize()
        ARnd2 = CInt(Rnd() * 3) 'Chooses a random number between 0 and 3 and assangs
it to the second integer variable

        tempAns = ARanArray(ARnd1) 'The string variable now contains the a part of
the array thats a value of rnd1 along
        ARanArray(ARnd1) = ARanArray(ARnd2) 'The value that was contained in that
part of the array is now equal to the value that was in the array rnd2 along
        ARanArray(ARnd2) = tempAns 'Now the value of the original rn1 is now placed
into the array rnd2 along
        'It has now swapped the two numbers around within the array
        Next
        'This loop therefore swaps different numbers 10 times essentially a shuffling the
array

        Rbtn1.Text = dtLdQtn.Rows(ranNum)(ARanArray(0))
        RBtn2.Text = dtLdQtn.Rows(ranNum)(ARanArray(1))
        RBtn3.Text = dtLdQtn.Rows(ranNum)(ARanArray(2))
        RBtn4.Text = dtLdQtn.Rows(ranNum)(ARanArray(3))
        'Loads all the answers within the array and equals them to the radio button
text's

        Details.QtnTotal += 1
        QNumLbl.Text = CStr(Details.QtnTotal)
        FeedBackLd()
    End Sub

    Private Sub AnsBtn_Click(sender As Object, e As EventArgs) Handles AnsBtn.Click

        Dim dtLdAns As New DataTable

        If Rbtn1.Checked Then 'If the 1st radio button is ticked then
            dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & Rbtn1.Text & "'")
        'Loads from the database the rows which have the PA1 value equal to the radiobutton text
            If dtLdAns.Rows.Count > 0 Then 'If there is more than 0 rows left then it
outputs a message saying they are correct
                MsgBox("                               CORRECT!
                                         Onto The Next Question   ")
                Details.QtnRight += 1 'If this message box is displayed then the question
right variable increments
            Else
                MsgBox("                               INCORRECT!
                                         Onto The Next Question   ")
            End If
        End If

        If RBtn2.Checked Then 'Same but with the 2nd radio button
            dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & RBtn2.Text & "'")
            If dtLdAns.Rows.Count > 0 Then
                MsgBox("                               CORRECT!
                                         Onto The Next Question   ")
            End If
        End If
    End Sub

```

```

        Details.QtnRight += 1
    Else
        MsgBox("INCORRECT!"

            Onto The Next Question  ")
    End If
End If

If RBtn3.Checked Then 'Same but with the 3rd radio button
    dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & RBtn3.Text & "'")
    If dtLdAns.Rows.Count > 0 Then
        MsgBox("CORRECT!

            Onto The Next Question  ")
        Details.QtnRight += 1
    Else
        MsgBox("INCORRECT!

            Onto The Next Question  ")
    End If
End If

If RBtn4.Checked Then 'Same but with the 4th radio button
    dtLdAns = runSQL("Select * from PQuestions where PA1 = '" & RBtn4.Text & "'")
    If dtLdAns.Rows.Count > 0 Then
        MsgBox("CORRECT!

            Onto The Next Question  ")
        Details.QtnRight += 1
    Else
        MsgBox("INCORRECT!

            Onto The Next Question  ")
    End If
End If

ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows ucLoad to
access AddUC
    ucLoad.addUC("ucPhysicsQuiz") 'Puts this string into the subroutine AddUC which
determines which user control to load
    Me.Dispose() 'Gets rid of the user control
    GC.Collect() 'Gather all the remaining data left after closing the user control

End Sub

Private Sub FeedBackLd()
    If Details.QtnTotal = 21 Then
        ucLoad = Me.Parent 'Sets ucLoad to the parent of main form which allows
ucLoad to access AddUC
        ucLoad.addUC("ucPhysicsFeedback") 'Puts this string into the subroutine AddUC
which determines which user control to load
        Me.Dispose() 'Gets rid of the user control
        GC.Collect() 'Gather all the remaining data left after closing the user
control
    End If
End Sub
End Class

```