T-LEVELS | Institute for Apprenticeships & Technical Education

Lead Examiner Report

Summer 2022

T Level Technical Qualification in Digital Production, Design and Development- Occupational Specialism

Pearson

Introduction

This was the first complete assessed series of the Occupational Specialism that took place.

The tasks set out in the assessment followed the format identified in both the Sample Assessment Materials (SAM) and the Additional Sample Assessment Materials (ADSAM) published on our website.

This assessment consisted of four tasks, requiring students to demonstrate knowledge of a range of specification topics and apply this knowledge to the scenario.

Individual questions

The following section considers each question on the paper, providing examples of popular student responses and a brief commentary of why the responses gained the marks they did. This section should be considered with the live external assessment and corresponding mark scheme.

Task 1- Activity A (ii) The Proposal

In the digital Distinction portfolio, the learner needs to consider the solution and provide a comprehensive proposal with the justification.

The proposal should clearly illustrate relevance to the scenario. It should not explain what functional requirements are or define the legislation. Decomposition should show the problem to be solved and how it meets the client and users needs. The risk associated needs to indicate what they are and how they will be mitigate.

The example below demonstrates some of these focus points and the traits associated with a Distinction grade.

Functional and non-functional requirements

Functional	Non-functional
1. To provide interactive teaching and learning resources in a range of subjects. a. Tests on various subjects b. English revision c. Maths's revision d. Science revision e. Computer Science revision 2. To provide access to digital content to encourage wider learning. a. Subject video links (YouTube) b. Tests/Quizzes c. Revision material 3. Support assessment and monitoring of learner progress. a. User accounts b. Database where progress is stored 4. To provide collaborative teaching and learning tools. a. Setting assignments for students 5. accessibility features to support a wide range of users. a. Voiced narration of page b. Colour contrast c. Light/dark mode d. Keyboard accessibility e. Alternative text 6. A learning reward system a. Progress linked to account b. Prizes/Rewards for learning c. Page which tracks your progress 7. Gamified learning. a. Word search for keywords b. Link to Kahoot on the subject 8. To ensure the solution meets the relevant regulatory guidelines and legal requirements.	1. Security considerations a. Strong password guidance b. Encryption c. Two-factor authentication 2. Required accessibility features a. Voiced narration of page b. Colour contrast c. Light/dark mode d. Keyboard accessibility e. Alternative text 3. Scalability requirements a. Layouts on different devices 4. Key performance indicators and metrics in relation to: a. Responsiveness b. Load handling c. Reliability 5. User acceptance criteria

The learner considered broader issues related to the scenario, the learner has functional and non-functional requirements, and it addresses the client needs and hence related to the scenario. They have also added justification for each requirement.

Functional requirements definitions

1 – Interactive teaching refers to content that the user can interact with in order to increase engagement when learning a subject, the "learning resources" refers to the many different types of media used by the digital solutions users in order to gain knowledge form a particular subject.

2 – Digital content refers to the many different types of media used by the digital solutions users in order to gain knowledge form a particular subject. Examples include: Textbooks, video content, gamified learning (Crosswords, mini games) and diagrams.

3 – Assessments refer to quizzes that the user can partake in to test their knowledge in various subjects. The tracking of user progress refers to the use of a database to record test scores and answers to different questions.

4 – Collaborative teaching and learning tools refer to digital tools which allow users to be able to interact with their peers/teachers through direct messaging, video calling or online group tasks. In the context of the digital solution this would be used to set assignments.

5 – Accessibility features refer to settings which can be turned on in order to increase the user experience of those with disabilities. An example would be subtitles on an educational video for someone who has difficulty hearing.

6 – A reward system could refer to any system that rewards its users for performing an action on the digital solution.

7 – Gamified learning refers to the creative way in which a learning environment is turned into a game which users can engage in whilst still learning new information as the 'game' requires the user to learn to play.

8 - Regulatory guidelines and legal requirements simply refers to the laws and industry specific guidelines such as The Data Protection Act and the Copyright, Designs and Patents Act which must not be broken otherwise the solution would be deemed illegal.

User acceptance criteria

User acceptance criteria refers to conditions that the proposed digital solution must satisfy to be accepted by a user. Acceptance criteria is helpful as it streamlines the acceptance testing process. The User acceptance criteria for GibJohn Tutoring's userbase are that:

1. If the user wishes to register for an account but their details have an incorrect syntax, or their password isn't deemed strong enough. An error message will appear informing the user of the actions that are needed for their details to be accepted.

2. If the user forgets their accounts password - Given the user is on the login page when the user selects the "Forgot password" button, then the user will be sent the password via their email account.

3. If the user wants to retry a test they have just completed to try and get a better score, when the user presses the "Try again" button the system should restart the test straight from the beginning.

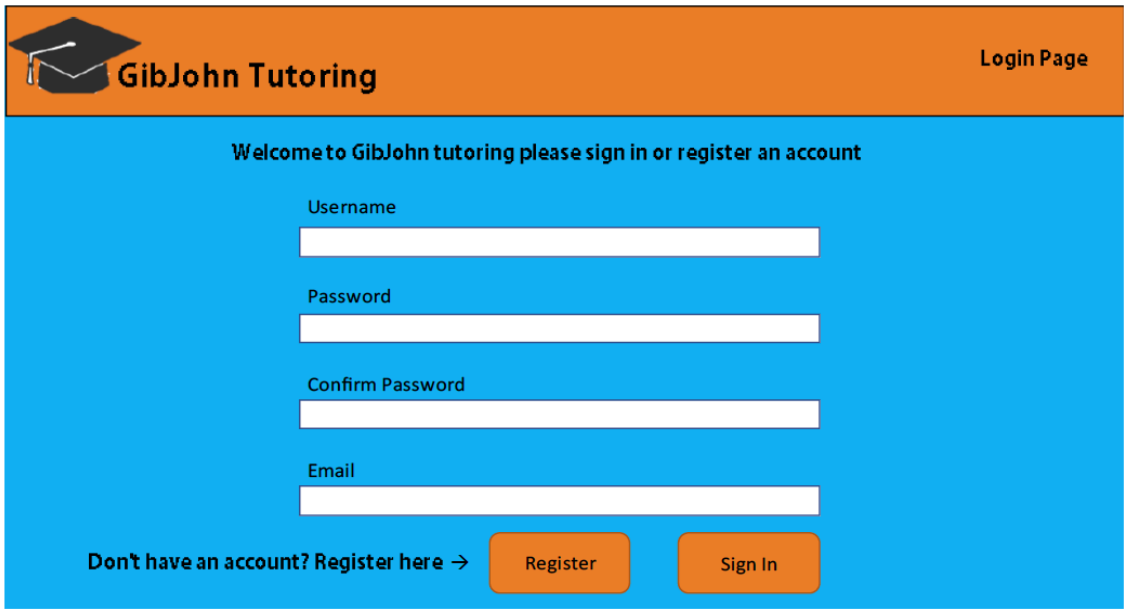
4. Given that the user registers for an account when the user returns to the dashboard then the user should now be signed into their account without having to go through a log in screen.

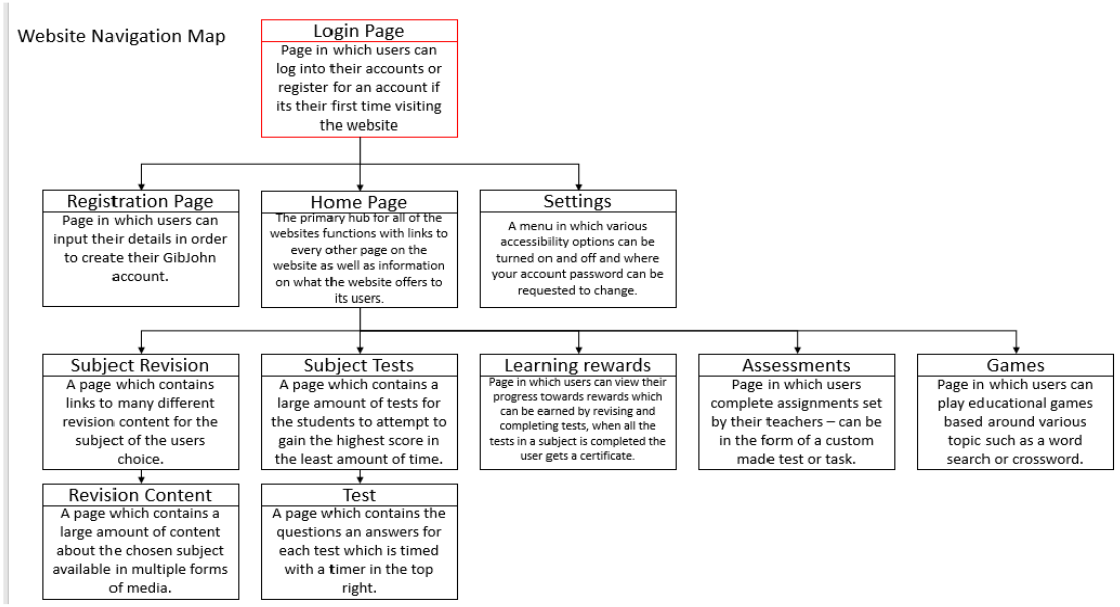
5. Given the users turns on an accessibility option in the settings, when the user returns to the dashboard/homepage then the changes should go into effect for example switching the forum from "light mode" to "dark mode".

6. As a student user, you should not be allowed to access other students progress pages, that privilege should be reserved for teachers, this is to ensure each students private information is kept between themselves and the teachers only.

For the 'User acceptable criteria' the learner has considered sensible criteria and justification of how it applies to the importance of how the solution meets the project objectives.

Visual/Interface designs





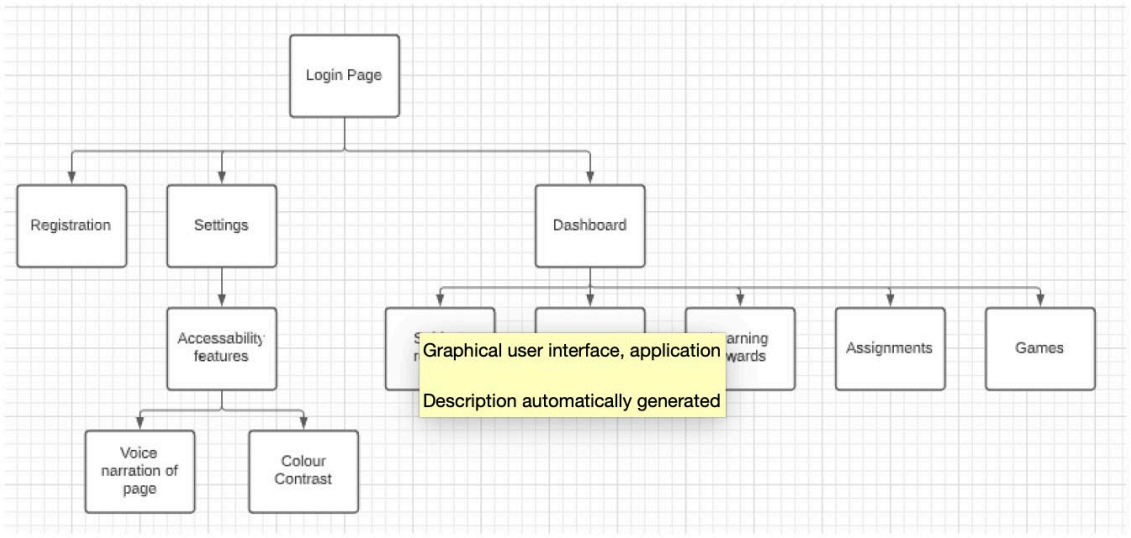
Page Specification:

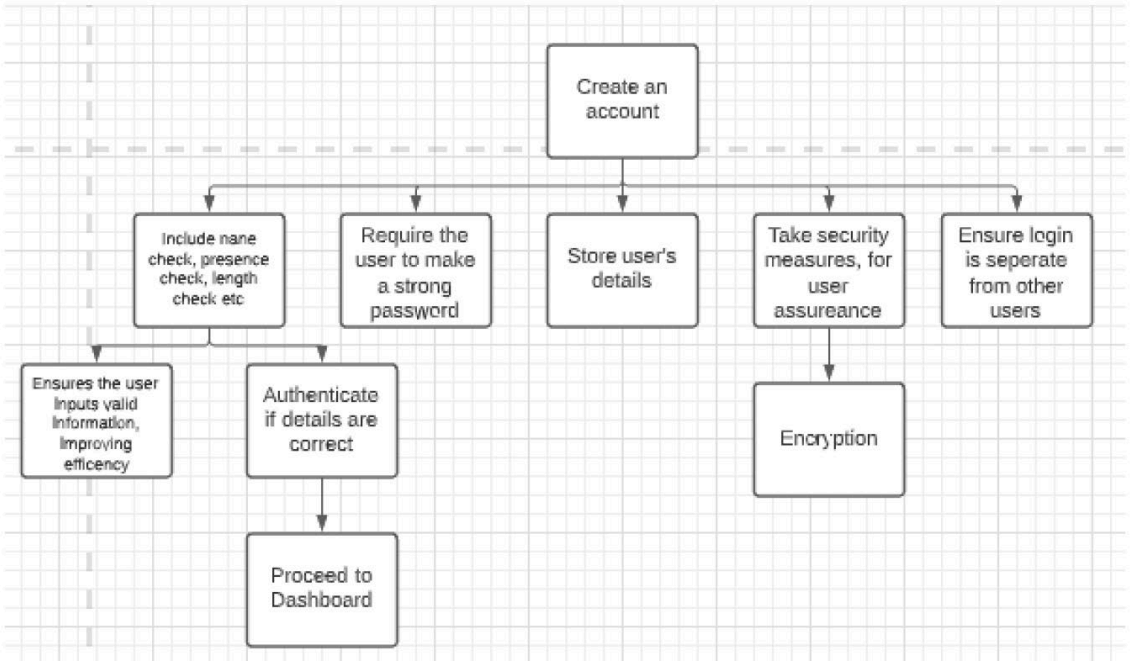
Interface

- Labels (text only)
- Bright, eye-catching colours (Light Blue and Orange Accent 2)
- Company Logo for brand recognition
- Font type Adobe Gothic Std B for professional look
- Minimalist design to not overload the user

The learner has provided evidence of detailed, compelling designs and shows enough for a third party to be able to create the whole artefact. The learner has also considered the complex structure of the website and page specifications such as background colour, text sizes and type. You will notice that the learner has provided layouts, visual hierarchy, and standard convention.

Learners that did not perform in this section generally had poor visual designs with little consideration of the typical conventions. For example, how would the Web developer build the website without knowing text sizes, colours and other descriptive features.





The learner has demonstrated their ability to decompose the problem into smaller sub-systems shown as a visual diagram; using stepwise refinement or sensible other notions is also acceptable. First, the learners must show the overall breakdown of the problem.

From here, they should select one sub-system and break it down in the next stage, which would be the algorithm using a flowchart or pseudocode as demonstrated below.

Algorithms

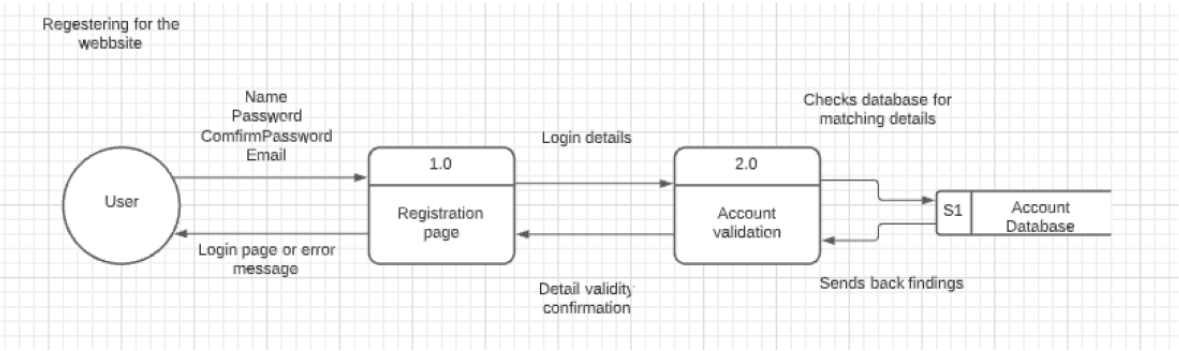
Front End Needs

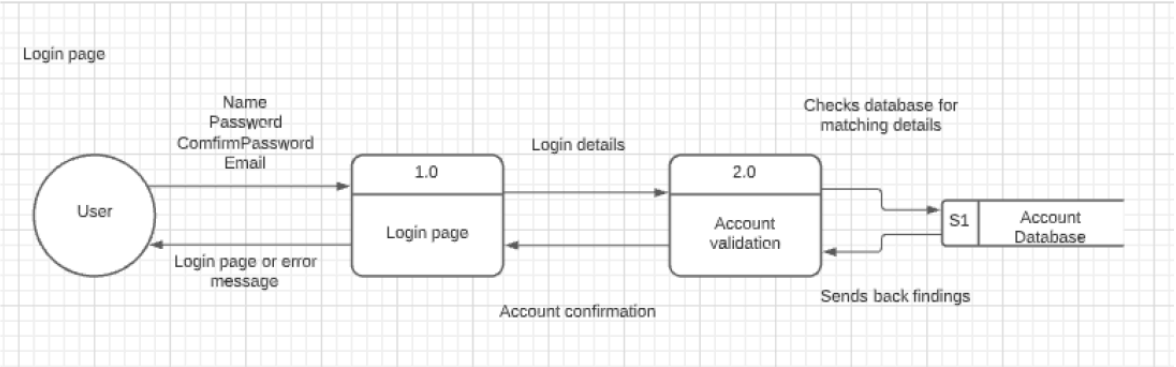
1. Clear buttons that are user friendly to use
2. Aesthetically pleasing User Interface design with simple layout
3. Clear links to the back end of the program – to ensure data the user inputs can be received.
4. Restricted user access to backend systems

Back End Needs

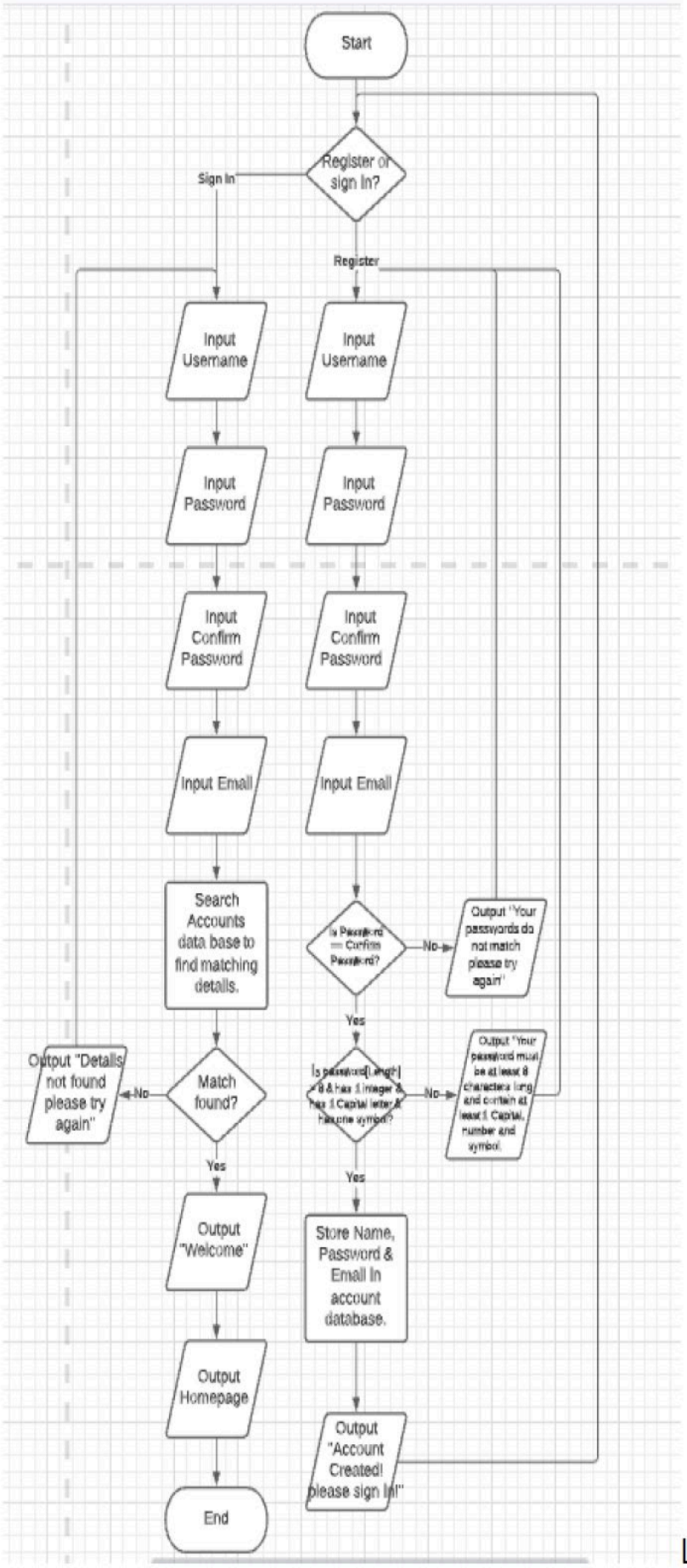
1. Data Storage (Database)
2. Data stored efficiently using tables to sort out data
3. The ability to read and write to and from the back end
4. Data Security such as encryption to protect users' personal data, adhering to the Data Protection Act
5. Link to the front end (Windows Forms C#) from back end (SQL)
6. Link between different queries and tables

Data Flow Diagrams





Flowcharts



Login/Register

Pseudocode

```
FUNCTION SignIn()
    OPEN Account_Database
    RECEIVE Name FROM (STRING) KEYBOARD
    RECEIVE Password FROM (STRING) KEYBOARD
    RECEIVE ConfirmPassword FROM (STRING) KEYBOARD
    RECEIVE Email FROM (STRING) KEYBOARD
    IF Password == ConfirmPassword THEN
        FOR each line in Account_Database
            READ Account_Database
            IF Name AND Password AND Email == Account_Database (Name, Password, Email) THEI
                SEND "Welcome" TO DISPLAY
            ELSE
                SEND "Incorrect details please try again." TO DISPLAY
    ELSE
        SEND "Your passwords do not match please try again" TO DISPLAY
END FUNCTION


FUNCTION CreateAccount()
    OPEN Account_Database
    SET Valid TO FALSE
    WHILE valid NOT TRUE
        RECEIVE Name FROM (STRING) KEYBOARD
        RECEIVE Password FROM (STRING) KEYBOARD
        RECEIVE ConfirmPassword FROM (STRING) KEYBOARD
        RECEIVE Email FROM (STRING) KEYBOARD
        IF Password == ConfirmPassword AND contains at least 1 Uppercase character, 1 lowercas
character, 1 integer and symbol AND Password(length) >= 8 THEN
            SET Valid TO TRUE
            WRITE Name, Password, Email TO Account_Database
            CALL FUNCTION SignIn()
        ELSE
```

Test method	White box testing
Purpose	The purpose of White box testing is to test the structure of the program, rather than the program's function. The program is analysed in a way such that all possible pathways that a user can take is tested to ensure the program functions as intended. This makes testing much more streamlined.
Testing data	All possible data inputs from the user are tested using every data type. Examples include: Valid data, Invalid data, Erroneous data, *null* data. Outputs as well as any errors will be recorded.
Test criteria	The output of the program should produce the expected outcome as I intended. Each individual test criteria will depend on the type of test being conducted. For example - Smoke testing will need any result to meet the criteria whilst unit testing will be specific to the function being tested.
When will this testing be used?	White box testing will be the most common testing method I use when testing my digital solution as the vast majority of tests will require the source code to be visible to see if the program is working as intended.
Test outcomes	The actual outcome data will be compared to the expected outcome and if the output differs from the expected outcome, then the issue and necessary changes will be added to the testing log.

Test method	Black box testing
Purpose	The purpose of Black Box Testing is to test the program against external factors responsible for software failures. Whilst white box testing

	focuses on the code, black box testing just focuses on the Input's and outputs of the system. Black Box testing is especially useful as it can be performed by users without any programming knowledge.
Testing data	The test data consists of the chosen input and output of the solution. Unlike white box testing it does not methodically check every possible input into the program.
Test criteria	The output of the program should produce the expected outcome as I intended. Each

The learner has explained the test strategy for the testing phase, in which they have to consider a comprehensive range of tests such as Whitebox and Blackbox testing. It should be regarded as to which test is used at what stage of the development of the artefact; the learner prepares this. The section is not a test plan that will follow in a later section when

the test strategy is carried out.

Similarly, lower-marked students tended to create test plans without any consideration of the strategy which led to issues such as test plans being developed after testing had been taken.

Weaker examples showed little consideration of the impact of any decisions that might have been made with apparent omissions of any contextual data.

Task 2

In this area, students can demonstrate their understanding by providing reasoned justifications for the approaches used. This could include considerations such as their approach to a solution. They must use two different programming languages. The most popular languages used for this series were Python, PHP, JS and SQL:, which is why specific server option were considered.

Justifications and depth are essential, so not only writing the code on multiple parts of the project; this is where learners can justify the decisions by adding comments on the code. It was also advantageous if the completed solution was either saved package ready to execute or students could provide a video which must be narrated.

Regarding the legal and regulatory guidelines and standards, the learners need to demonstrate how the website would be viewed in different browsers. They can validate the code on the W3C website. This will let the students know what needs to be added for the website to be viewed as expected, no matter which browser.

Security controls we are looking for consideration of using username and password, cookie notification, and even two-way authentication. This provides evidence for legal and ethical consideration.

Very weak examples tended to describe what had taken place instead of providing clear justifications for approaches taken. They tended not to add comments. The code was linear in structure with lots of errors and was a partially working solution.

NEW CODE (Screenshot 44)

```
bool matchFound = false;     // Sets the value of matchFound to false to ensure a value is returned if a mat

MySqlConnection Gj_connect = ConnectToDB();  // Sets the Connect To Database Command to a variable.

MySqlDataReader myDataReader;               // Sets the SQL Data reader function to be called using a variab

string CommandText = "SELECT Username, Password, Email FROM user_details WHERE" +  // Selects the users deta
    " Username=@name AND Password=@passw";

using (Gj_connect)
{
    MySqlCommand findUser = new MySqlCommand(CommandText, Gj_connect);

    findUser.Parameters.Add("@name", MySqlDbType.VarChar);    // Sets the parameters for the findUser comman
    findUser.Parameters["@name"].Value = txt_Username.Text;

    findUser.Parameters.Add("@passw", MySqlDbType.VarChar);
    findUser.Parameters["@passw"].Value = txt_Password.Text;

    try
    {
        Gj_connect.Open();

        myDataReader = findUser.ExecuteReader();     // Executes the findUser Command.

        if (myDataReader.Read())  // If there is any data to read...
        {
            matchFound = true;   // Sets the value of matchFound to true.
            // Displays the user details for confirmation and testing purposes.
            string name = myDataReader.GetString(0);
            string email = myDataReader["Email"].ToString();
            string pass = myDataReader.GetString(2);

            MessageBox.Show(string.Format("Username: {0}\nPassword: {1}\nEmail: {2}", name, pass, email));
        }
```

Screenshot 45

OLD CODE (Screenshot 41)

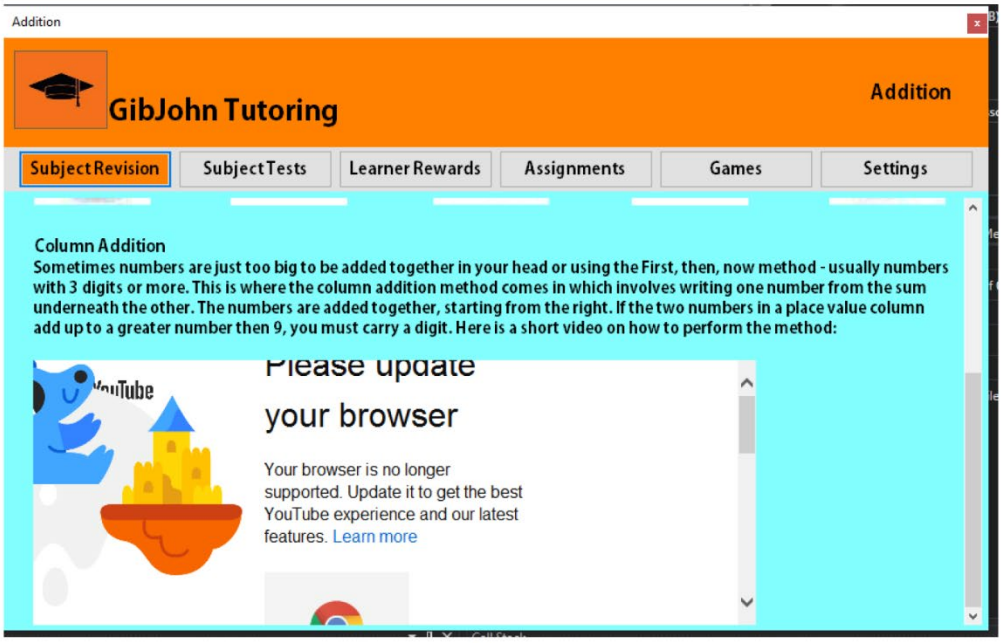
```
string CommandText = "SELECT Username, Password, Email FROM user_details WHERE" +
    "Username=@name AND Password=@passw;";
```

NEW CODE (Screenshot 42)

```
string CommandText = "SELECT Username, Password, Email FROM gibjohn_database.user_details WHERE" +
    "Username=@name AND Password=@passw;";
```

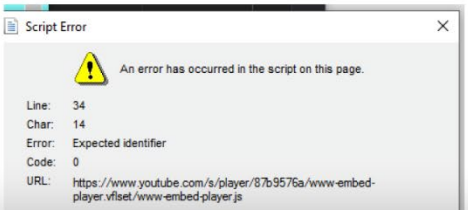
OLD CODE (Screenshot 43)

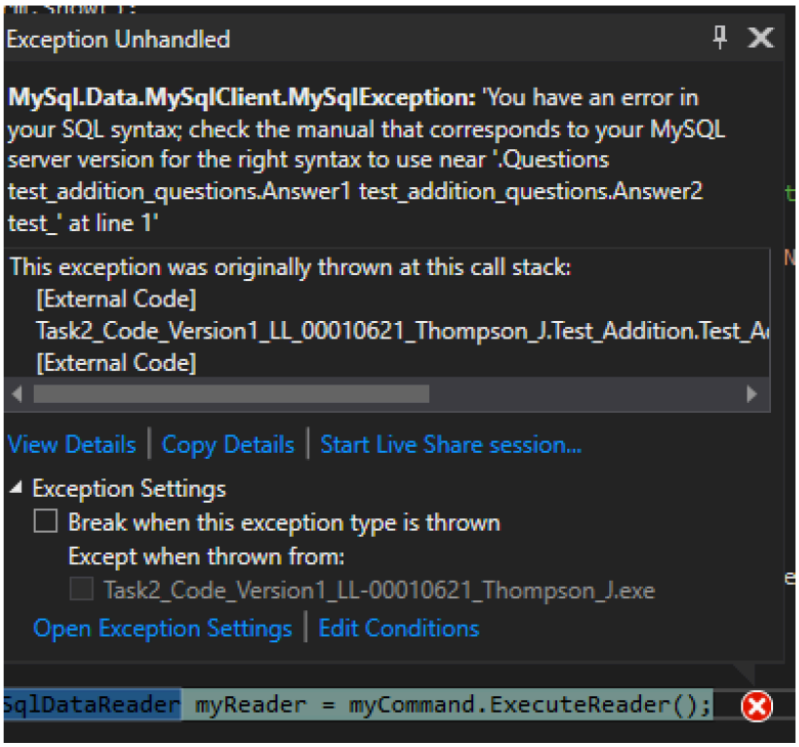
```
bool matchFound = false;     // Sets the value of matchFound to false to ensure a value is returned if a match cannot

MySqlConnection Gj_connect = ConnectToDB();  // Sets the Connect To Database Command to a variable.

MySqlDataReader myDataReader;               // Sets the SQL Data reader function to be called using a variable.

string CommandText = "SELECT Username, Password, Email FROM gibjohn_database.user_details WHERE" +  // Selects the u
    "Username=@name AND Password=@passw;";

using (Gj_connect)
{
    MySqlCommand findUser = new MySqlCommand(CommandText, Gj_connect);

    findUser.Parameters.Add("@name", MySqlDbType.VarChar);    // Sets the parameters for the findUser command.
    findUser.Parameters["@name"].Value = txt_Username.Text;

    findUser.Parameters.Add("@passw", MySqlDbType.VarChar);
    findUser.Parameters["@passw"].Value = txt_Password.Text;

    try
    {
        Gj_connect.Open();

        myDataReader = findUser.ExecuteReader();     // Executes the findUser Command.

        if (myDataReader.Read())  // If there is any data to read...
        {
            matchFound = true;   // Sets the value of matchFound to true.
            // Displays the user details for confirmation and testing purposes.
            string name = myDataReader.GetString(0);
            string email = myDataReader["Email"].ToString();
            string pass = myDataReader.GetString(2);

            MessageBox.Show(string.Format("Username: {0}\nPassword: {1}\nEmail: {2}\n", name, pass, email));
        }
```



To fix this error instead of using the "https://www.youtube.com/watch?v=v9J3VuI8_y0" URL I would use the "https://www.youtube.com/embed/v9J3VuI8_y0" URL instead.

After switching the URL to /embed/ when I ran the code, I got 3 different errors:





Screenshot 70

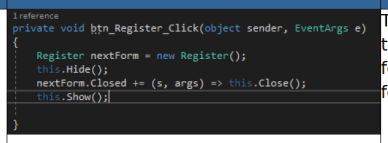
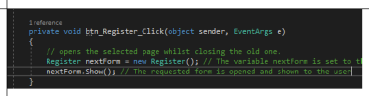
The learner has clearly demonstrated efficient functional code using two languages. We tested the solution as the learner provided executable files with 'read me file'. This helped focus point on user experience, and were able to test the input handling, error messages and outputs.

The student demonstrated well-organised code that includes naming conventions and comments that allow third parties to pick up on what the code is doing. It is logical and precise.

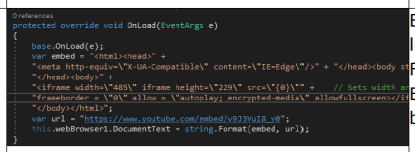
Lower marks were awarded for lack of detail or no comments, poorly organized code, and little evidence of a working solution. In addition, it lacks any consideration for standards and guidelines concerning accessibility and compatibility.

Task 2: Test log (Evidence located at end of table)

All tests will be conducted using White Box Testing:

escription of test	Test data to be used (if required)	Expected outcome	Actual outcome	Comments and intended actions
gister" button will pressed. (Normal a)		The program will open the registration page form and close the login form.	The login page was hidden, closed and then opened again. (Screenshot 1)	I believe the issue is that I accidentally put "this.Show()" referring to the current form rather then the variable "nextForm". I intend to correct this and try again.
gister" button will pressed. (Normal a)		The program will open the registration page form and close the login form.	The register page was opened successfully. (Screenshot 2)	Whilst the test was successful I believe it would be best to have the first form stay open as otherwise there would be no way to get back to the login page.
gister" button will pressed. (normal a)		The program will open the registration page form.	The register page was opened successfully. (Screenshot 3)	I am happy with the results and no more tests are required.

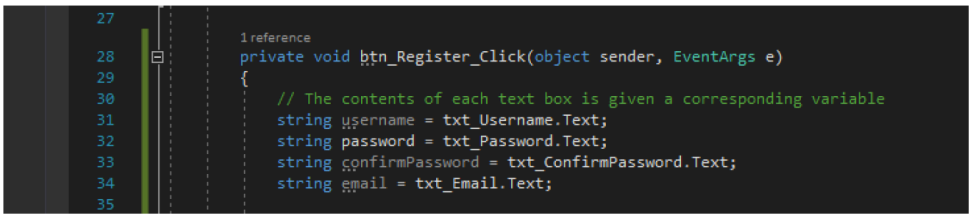
ting all buttons on 14 forms to see if y all open the rect form when ssed (normal data)	[Name of Form] nextForm = new [Name of Form]();
	this.Hide();
	nextForm.Closed += (s, args) => this.Close();
		All buttons are functional and work as intended.	All buttons are functional and work as intended.	Due to the sheer number of individual test logs this would create, I have merged each individual test into one.

Increasing the height of the embedded video. (normal data)		Embedded Video will load in the "Addition" Form and its height will have been increased.	Embedded Video will load in the "Addition" Form and its height has been increased. (Screenshot 11)	My theory was correct, I am now planning to manually edit the width and height values to fit to the size of the WebBrowser box then change the web browsers background to black to give the page a more professional look.
Checking if the modifications to the videos background and height worked.		Embedded Video will load in the "Addition" Form fitting into the Web Browser and having a black background.	Embedded Video loaded in the "Addition" Form fitting into the Web Browser and having a black background perfectly. (Screenshot 12)	The video has been successfully added to the form and is aesthetically pleasing to the user.
Register Password validation test (valid data)	Password: Qa1234567! Confirm Password: Qa1234567!	Messagebox Displays "Valid details" to screen.	Messagebox Displays "Valid details" to screen. (Screenshot 13)	The normal data was accepted as I predicted which means each IF statement is working as intended.
Register Password validation test (Invalid data)	Password: QA1234567! Confirm Password: QA1234567!	Messagebox Displays "Password should contain at least one lower case letter" to screen.	Messagebox Displays "Password should contain at least one lower case letter" to screen. (Screenshot 14)	No actions needed.
Register Password validation test (Invalid data)	Password: qa1234567! Confirm Password: qa1234567!	Messagebox Displays "Password should contain at least one upper case letter" to screen.	Messagebox Displays "Password should contain at least one upper case letter" to screen. (Screenshot 15)	No actions needed.

Functional requirement 3

Creating/Signing into an account allowing for the monitoring of learner progress.

Next, I began to add functionality to my Login and Registration screens – allowing the user to actually "Log in" and "Register" to access the website's contents. For the login screen, this will be achieved by validating the user's entered details and checking them against the existing values in the SQL database. For the Registration screen, Regex will be used to validate the user's password and the username will be checked against the SQL database to ensure it is unique. The first form I worked on was the registration page. First, I set the users inputs to variables so that they could be referenced later in the program.



```
27
        1 reference
28      private void btn_Register_Click(object sender, EventArgs e)
29      {
30          // The contents of each text box is given a corresponding variable
31          string username = txt_Username.Text;
32          string password = txt_Password.Text;
33          string confirmPassword = txt_ConfirmPassword.Text;
34          string email = txt_Email.Text;
35
```

Then, by inserting Regex into my program, the user's password could be checked to see if it contains at least 1 Uppercase letter, 1 Lowercase letter, 1 number, 1 symbol and be between 8 and 20 characters long. If the password does not contain 1 of these requirements a message will be sent to the user to explain the error.

Iterative Testing Example

For my testing I have taken an iterative approach by tackling testing in a 4-step process:

1. First, the requirements for my program are analysed.

2. Next, I design my initial solution using C# and SQL.

3. After that, Implement and test the code to see if it functions as intended.

4. Finally, I review my work based on the test results and edit the program accordingly.

Here is an example:

Testing requires students to demonstrate their knowledge around testing and debugging code. To do this, the students use a test plan template and the program's requirements. Students should use these to debug and test the code and try to implement a functioning solution.

Weaker examples showed that students relied heavily on the IDE to identify syntactic errors but did not go beyond this.

Where more robust examples showed an awareness of the function of the code (testing inputs and returns) as well as the role that validations and verification might play in the function of the code, with a clear record of actions taken to fix any issues encountered.

The testing evidence was often mirrored in the attempts to resolve the issues to get the code running and make sure it met the user requirements.

Areas for consideration in this task moving forward-

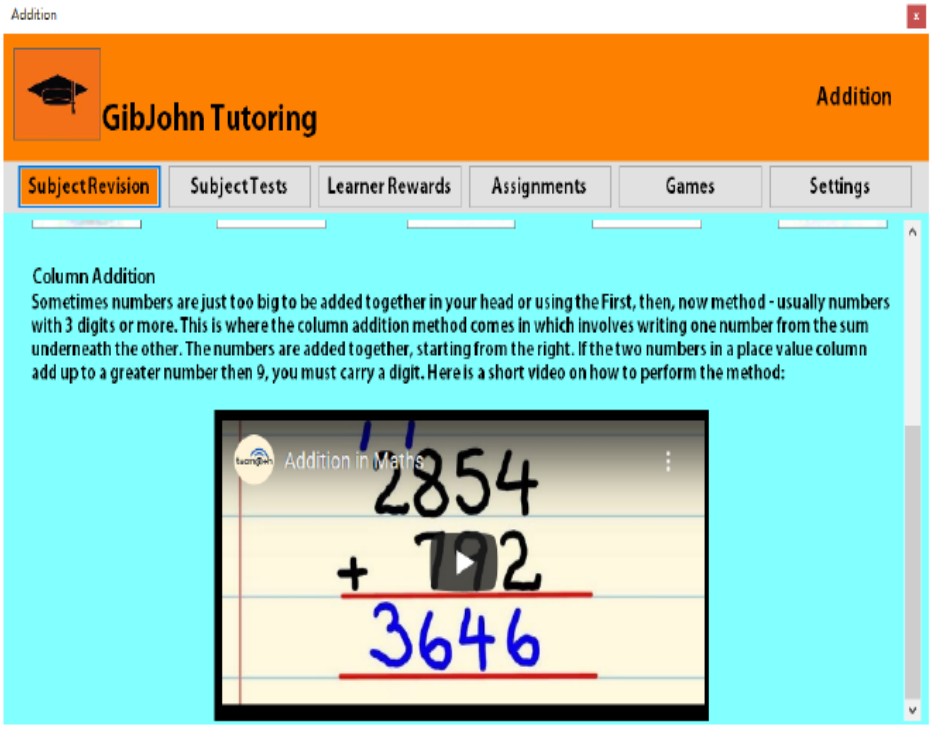
- Encourage students to consider the requirements of code as outlined in the task description, to help inform a testing strategy beyond syntax errors
- Encourage students to explore a range of methods of validation and or verification of inputs
- Encourage regression testing of any implemented solutions
- Encourage students to use a range of test data- normal, erroneous and extreme data

Task 3a

The learner must show two different techniques for gathering feedback, e.g., survey and observations, which help the development of the prototype. The question should be directed at technical and non-technical testers. The question should be appropriate so that data drawn from it can be used to benefit the artefact and the user. The learner should provide a detailed visualization of the data gathered and analysed. This offers the opportunity to develop the prototype further.

GibJohn Tutoring could replace the content taken from BBC Bitesize with their own teaching materials in a future iteration of the website, but the prototype displays the potential of what the website's revision pages could look like.

Finally, a YouTube video on addition has been linked to the revision page:



I believe this asset was effective as the video matched the topic being taught and offered the user access to digital content to encourage wider learning.

21. Did you encounter any bugs, errors or crashes when Logging in to GibJohn Tutoring?
Yes
No

22. On a scale of 1 - 10, How did you find the signing up process?
10 - Perfect
9 - Excellent
8 - Very Good
7 - Good
6 - Above Average
5 - Below Average
4 - Poor
3 - Very Poor
2 - Awful
1 - Non-Functioning

23. What part of the Signing/Logging in process did you find to work well?
The password validation

24. How could the Signing/Logging in process be improved?
When signing maybe change it so you do not have to input all your details Again

Homepage

Questions based around the homepage (Form 3)

25. Was the homepage helpful at presenting the websites content to the user?
Yes
No

Gathering feedback plan

My plan for gathering feedback for the GibJohn Tutoring website prototype is by creating an online questionnaire to fill out which targets both a technical (e.g. Programming professionals) and non-technical audience (e.g. the client, users).

The results of the survey will give me the data I need to begin improving the prototype to be closer to the final product. I will create this questionnaire using Microsoft Forms and giving the user supervised access to the program, documentation and testing logs. Non-Technical users will answer general questions about the program whilst technical users with answer more in-depth programming-based questions. The questions are designed to give both quantitative and qualitative feedback to help inform future development of the GibJohn Tutoring Website. Here is a simplified version of the plan:

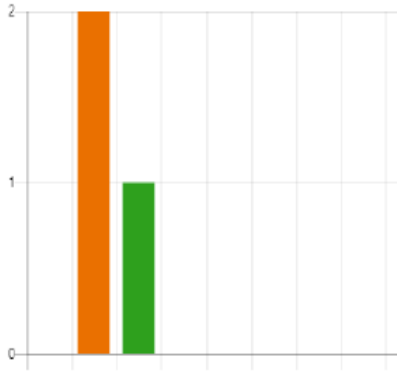
1. Create Questionnaire which can be answered by technical and non-technical users based on the GibJohn Tutoring website prototype.

2. Gather feedback using Questionnaire receiving quantitative and qualitative answers.

3. Use feedback data to inform future development of the prototype website.

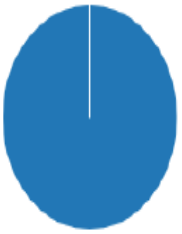
4. On a scale of 1 - 10, how would you rate the prototypes programming?

10 - Perfect	0
9 - Excellent	2
8 - Very Good	1
7 - Good	0
6 - Above Average	0
5 - Below Average	0
4 - Poor	0
3 - Very Poor	0
2 - Awful	0
1 - Non-Functioning	0



5. Are the code comments provided helpful to the average user?

Yes	3
No	0



6. If No - How would you improve the comments to be more useful?

Task 3b

Learners should provide the assets used and how valid the sources are, ensuring that no copyright laws have been broken and the reason as to why they chose them; this should be related to the scenario and why the object was used on the website page and how it would help the user.

Evaluations were appropriate and showed an understanding of the requirements of the set task brief. While learners typically at this level provide evaluative comments concerning the requirements of the task, at the borderline, user needs were not effectively considered.

The validity and reliability of the sources of information used

The sources in which I got the written content for the 'Addition' revision page was from the website BBC Bitesize which is a very reliable source as it is a well-known educational site funded by the BBC. The content is valid as it has been produced by people in the education industry such as teachers and subject professionals.

The snippet of code I used to be able to link a YouTube video to visual studio, was taken from a post on the Stack Overflow website which is a website designed for programmers to ask questions and exchange solutions with one another. This content is valid as the code I incorporated into my program worked as intended and the Stack Overflow website is reliable as it is tightly moderated and designed to share source code.

```
79          }
            0 references
80          protected override void OnLoad(EventArgs e)
81          {
82              base.OnLoad(e);
83              var embed = "<html><head>" +
84              "<meta http-equiv=\"X-UA-Compatible\" content=\"IE=Edge\"/>" +
85              "</head><body>" +
86              "<iframe width=\"300\" src=\"{0}\"" +
87              "frameborder = \"0\" allow = \"autoplay; encrypted-media\" allowfullscreen></iframe>" +
88              "</body></html>";
89              var url = "https://www.youtube.com/embed/v0J3VuI8_y0";
90              this.webBrowser1.DocumentText = string.Format(embed, url);
91          }
92      }
93  }
94
```

Any legal and ethical implications of the assets and content selected

When choosing the assets I would use for the prototype, I paid close attention to only use uncopyrighted images / material in order to adhere to the Copyright and Patents Act.

Re-using BBC Bitesize's content on addition revision could potentially be seen as unethical however the purpose of using that asset was to have a temporary view on how the final product would look like with GibJohn Tutoring's revision content – not to steal others content and flog it off as their own. It was to demonstrate the proof-of-concept and will be updated in a later iteration.

Using the program code of a user on Stack Overflow is both legal and ethical as the entire purpose of the website is to share solutions on programming challenges other people are facing. All program code on the website is meant to be shared with other people. In addition, the code that was taken, was edited to better suit the program's purpose.