Higher
Coursework
Assessment Task

Higher Computing Science
Assignment
Assessment task

This document provides information for teachers and lecturers about the coursework component of this course in terms of the skills, knowledge and understanding that are assessed. It must be read in conjunction with the course specification.

Valid for session 2022-23 only.

This assessment is given to centres in strictest confidence. You must keep it in a secure place until it is used.

This edition: January 2023 (version 1.0)

© Scottish Qualifications Authority 2023

Contents

Introduction 3

Instructions for teachers and lecturers 4

Instructions for candidates 9

Introduction

This document contains instructions for teachers and lecturers, and instructions for candidates for the Higher Computing Science assignment. You must read it in conjunction with the course specification.

This assignment has 40 marks out of a total of 120 marks available for the course assessment.

This is one of two course assessment components. The other component is a question paper.

Instructions for teachers and lecturers

This assessment applies to the assignment for Higher Computing Science for the academic session 2022–23.

The task is valid for 2022–23 only. Once complete, you must send the assignment responses to SQA to be marked.

You must conduct the assignment under a high degree of supervision and control. This means:

- candidates must be supervised throughout the session(s)
- candidates must not have access to email or mobile phones
- candidates must complete their work independently — no group work is permitted
- candidates must not interact with each other
- with no interruption for targeted learning and teaching
- in a classroom environment

Time

Candidates have 6 hours to carry out the assignment, starting at an appropriate point in the course, after all content has been delivered. It is not anticipated that this is a continuous 6-hour session, although it can be, but conducted over several shorter sessions. This is at your discretion.

You have a responsibility to manage candidates' work, distributing it at the beginning and collecting it in at the end of each session, and storing it securely in between. This activity does not count towards the total time permitted for candidates to complete the assignment.

Candidates are prompted to print their work at appropriate stages of the tasks. They can print on an ongoing basis or save their work and print it later. Whatever approach they take, time for printing is not part of the 6 hours permitted for the assignment.

Resources

Each candidate must have access to a computer system with a high-level (textual) programming language and either:

- a database application or software that can create, edit and run SQL
- software that can create, edit and run HTML, CSS and JavaScript

This is an open-book assessment. Candidates can access resources such as programming manuals, class notes, textbooks and programs they have written throughout the course. These may be online resources.

You must not create learning and teaching tasks that make use of constructs required in the assessment task, **with the specific purpose of developing a solution that candidates can access during the assignment.**

There may be instances where restriction of network use is prohibited (for example, a local authority-managed network with specific limitations). However, it remains your professional responsibility to make every effort to meet the assessment conditions.

Reasonable assistance

The assignment consists of three independent tasks. They are designed in a way that does not require you to provide support to candidates, other than to ensure that they have access to the necessary resources. Candidates can complete the tasks in any order.

Once the assignment is complete, you must not return it to the candidate for further work to improve their mark. You must not provide feedback to candidates or offer an opinion on the perceived quality or completeness of the assignment response, at any stage.

You can provide reasonable assistance to support candidates with the following aspects of their assignments:

- printing, collating and labelling their evidence to ensure it is in the format specified by SQA
- ensuring candidates have all the materials and equipment required to complete the assignment — this includes any files provided by SQA
- ensuring candidates understand the conditions of assessment and any administrative arrangements around the submission and storage of evidence, and the provision of files
- technical support

Evidence

All candidate evidence (whether created manually or electronically) must be submitted to SQA in a paper-based format. The evidence checklist details all evidence to be gathered. You can use it to ensure you submit all evidence to SQA.

You should advise candidates that evidence, especially code, must be clear and legible. This is particularly important when pasting screenshots into a document.

There is no need for evidence to be printed single sided or in colour.

Alteration or adaptation

The tasks are in PDF and Word formats. Each task is available as a separate file from the secure site. Word files allow candidates to word process their responses to parts of the task.

You must not adapt the assignment in any way that changes the instructions to the candidate and/or the nature and content of the tasks. However, you can make changes to font size, type and colour and to the size of diagrams for candidates with different assessment needs, for example, visual impairment.

If you are concerned that any particular adaptation changes the nature and/or the content of the task, please contact our Assessment Arrangements Team for advice as soon possible at aarequests@sqa.org.uk.

Submission

Each page for submission has the number of the assignment task that it refers to, for example 1a, and contains space for candidates to complete their name and candidate number. Any other pages submitted, for example, prints of program listings or screenshots, must have this information added to them.

Specific instructions for teachers and lecturers: 2022-23

All candidates must complete task 1 (software design and development) and **either** task 2 (database design and development) **or** task 3 (web design and development).

It is at your discretion how you approach this optionality in assessment. The task your candidates complete might be pre-determined by your progress through the course, or you may be able to let candidates choose which task to complete.

You must follow these specific instructions and ensure that candidates are aware of what you will give them at each stage in the assessment.

Print each task on single-sided paper, where applicable, as this:

- allows candidates to refer to information on other pages
- helps you manage tasks that are split into more than one part

Task 1 — part A requires candidates to analyse and design a solution to a software problem. They must submit their evidence to you before you issue part B.

Task 1 — part B is a separate section. This ensures that candidates do not access part A and change their responses. Candidates must still have access to the problem description page during part B.

A CSV file 'attractions.csv' is provided for candidates to use in this part of the task.

Task 2 — part A requires candidates to analyse a database. They must submit their evidence to you before you issue part B.

Task 2 — part B is a separate section. This ensures that candidates do not access part A and change their responses.

A Microsoft Access file (gnomeSales.accdb) is provided for candidates to use in part B.

If your centre uses a different database system, you can create the relational database for part B using the CSV or text files provided. The CSV files contain the data for each table. The text files contain SQL create and insert statements for each table. In both cases, you will have to add primary keys and foreign keys as specified below.

Specific instructions for database setup
The 'gnomeSales' database includes table names, field names, primary keys, and foreign keys.

You do not need to add validation to any of the fields in the database tables.

gnomeSales database			
Customer	**CustOrder**	**GnomePurchase**	**Gnome**
customerID	orderID	gnomePurchaseID	gnomeID
forename	orderDate	orderID*	gnomeName
surname	customerID*	gnomeID*	description
houseNumber		quantity	unitPrice
streetName			
postcode			
emailAddress			

Task 2d requires candidates to test an SQL statement. You must provide this to candidates as part of the database. The SQL statement is already included in the Access file. If you use a different database management system, you should use the supplied text file (Query for 2d.txt) to add it to the database you provide to candidates.

Task 3 — part A requires candidates to analyse a website. They must submit their evidence to you before you issue part B.

Task 3 — part B is a separate section. This ensures that candidates do not access part A and change their responses. They must submit their evidence for part B to you before you issue part C.

Task 3 — part C is a separate section. This ensures that candidates do not access part B and change their responses.

A folder Web files has been provided. This contains the CSS, HTML and the image files candidates need to complete this task. These files must not be renamed and they must remain in the folders provided.

Candidates **do not** need to print completed web pages in colour.

Instructions for candidates

This assessment applies to the assignment for Higher Computing Science.

This assignment has 40 marks out of a total of 120 marks available for the course assessment.

It assesses the following skills, knowledge and understanding:

- applying aspects of computational thinking across a range of contexts
- analysing problems within computing science across a range of contemporary contexts
- designing, implementing, testing and evaluating digital solutions (including computer programs) to problems across a range of contemporary contexts
- demonstrating skills in computer programming
- applying computing science concepts and techniques to create solutions across a range of contexts

Your teacher or lecturer will let you know if there are any specific conditions for doing this assessment.

In this assessment, you have to complete **two** short practical tasks.

You must complete task 1 (software design and development) and **either** task 2 (database design and development) **or** task 3 (web design and development).

You may complete the tasks in any order.

Advice on how to plan your time

You have 6 hours to complete the assignment. Marks are allocated as follows:

- Task 1 — software design and development 25 marks (63% of total)

AND EITHER

- Task 2 — database design and development 15 marks (37% of total)

OR

- Task 3 — web design and development 15 marks (37% of total)

You can use this split as a guide when planning your time for each of the two tasks.

Advice on gathering evidence

As you complete each task, you must gather evidence as instructed in each task.

Your evidence, especially code, must be clear and legible. This is particularly important when you paste screenshots into a document.

Use the evidence checklist provided to make sure you submit everything necessary at the end of the assignment. Ensure your name and candidate number is included on all your evidence.

Evidence may take the form of printouts of code/screenshots/typed answers, hand-written answers or drawings of diagrams/designs.

Advice on assistance

This is an open-book assessment. This means that you can use:

- any classroom resource as a form of reference (for example programming manuals, class notes, and textbooks) — these may be online resources
- any files you have previously created throughout the course

The tasks are designed so you can complete them independently, without any support from your teacher or lecturer. This means that you:

- cannot ask how to complete any of the tasks
- cannot access any assignment files outside the classroom

Computing Science assessment task: evidence checklist

You should complete the checklist for task 1 and **either** task 2 or task 3.

Task 1		
Part A evidence		
1a	Completed task sheet identifying processes for the program	☐
1b	Completed task sheet showing completed data flow design	☐
Part B evidence		
1c (i) and (ii)	Printout of your completed program code	☐
	Printout of program outputs	☐
	Printout of service.csv file	☐
1d	Completed task sheet showing completed trace table	☐
1e	Completed task sheet evaluating maintainability	☐

Task 2		
Part A evidence		
2a	Completed task sheet identifying two functional requirements of the database	☐
Part B evidence		
2b	Printout of SQL statement	☐
	Printout of the output produced	☐
2c	Printout of SQL statement(s)	☐
	Printout of the output produced	☐
2d	Printout of the amended SQL statement	☐
	Printout of the output produced	☐
2e	Complete task sheet evaluating the accuracy of output	☐

Task 3		
Part A evidence		
3a	Completed task sheet with two functional requirements for the website	☐
Part B evidence		
3b	Completed task sheet showing the completed wireframe	☐
Part C evidence		
3c	Printout of edited 'reviews.html' file	☐
	Printout of the 'Reviews' page, as viewed in a browser	☐
3d	Printout of edited 'submitReview.html' file and 'styles.css' file showing code	☐
	Printout of the 'Submit Review' page, as viewed in a browser	☐
3e	Completed task sheet describing how you would test form elements	☐
3f	Completed task sheet evaluating the fitness for purpose of the website	☐

Please follow the steps below before handing your evidence to your teacher or lecturer:

- Check you have completed all parts of tasks 1, and one of **either** task 2 or task 3.
- Label any printouts/screenshots with the task number (for example 1a, 2a).
- Clearly display your name and candidate number on each printout.

Task 1: software design and development

Problem description

The manager of a theme park would like a program to provide him with data on the park's 26 attractions.

Purpose

The program is required to read the attraction data from a CSV file into parallel arrays. The file contains the attraction name, category of attraction, total number of visitors, number of days open and the height restriction.

The program will find the names of the least visited attraction(s) and the most visited attraction(s).

Attractions in the category 'Roller Coaster' are serviced every 90 days. The program is required to calculate the number of days until the next service. It will use this information to produce an output that identifies all roller coasters that need to be serviced within the next 7 days.

Assumptions

- the CSV data file is formatted correctly, is error-free and is updated every day

An example of the data in the file is shown below:

```
Aftershock,Roller Coaster,510324,695,1.2m
Aqua Loop,Water,157288,542,0.9m
Asteroid Belt,Roller Coaster,551218,623,1.4m
Attack of the Smartphones,Simulation,548630,663,1.0m
Beaver Falls,Water,95970,695,1.2m
Bug Blaster,Simulation,293033,542,0.9m
Bumblebee Flyer,Kids,104010,715,0.9m
Candyfloss Carousel,Kids,95970,663,0.9m
...
```

Task 1: software design and development (part A)

1a One process for the program is stated below.

Using the problem description and purpose, identify the remaining processes for the program.

(3 marks)

Find the name(s) of the attraction(s) with the lowest number of visitors.

Candidate name_____ Candidate number_____

1b The top level design for the program is shown below.

Complete the design to show the missing data flow for each step.

(2 marks)

Top level design		
Read data from file into parallel arrays	IN	
	OUT	attraction(), category(), visitors(), daysOpen(), height()
Find and display the names of the least visited and most visited attractions	IN	
	OUT	
Write to file the names of roller coasters that need a service within 7 days	IN	
	OUT	

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

Task 1: software design and development (part B)

Your teacher or lecturer will provide you with a file called 'attractions.csv'.

The design for the program is shown below.

1	Read data from file into parallel arrays	OUT	attraction(), category(), visitors(), daysOpen(), height()
2	Find and display the names of the least visited and most visited attractions	IN	attraction(), visitors()
3	Write to file the names of roller coasters that need a service within 7 days	IN	attraction(), category(), daysOpen()

Refinements

```
3.1  Create 'service.csv' file
3.2  Loop for each attraction
3.3      If current category is 'Roller Coaster' then
3.4          Set days to current daysOpen modulus 90
3.5          If (90 – days) is less than or equal to 7 then
3.6              Write current attraction to file
3.7          End if
3.8      End if
3.9  End loop
3.10 Close 'service.csv' file
```

1c(i) Using the problem description and design, implement the program in a language of your choice.

Your program should:

- read data from the 'attractions.csv' file and store into parallel arrays
- find and display the name(s) of the:
  - least visited attraction(s)
  - most visited attraction(s)
- write the names of roller coasters that require a service within 7 days to 'service.csv'
- be maintainable and modular

(12 marks)

1c(ii) The theme park's manager wants an update on attractions with a height restriction of 1.0m and above. Height restrictions range from 0.9m to 1.4m.

Implement a new sub-program to count and display the number of attractions with a height restriction beginning with the character '1'.

(3 marks)

Print evidence of:

- your program code
- program outputs from 1c(i) and 1c(ii)
- the 'service.csv' file

1d The sub-program 'Write to file the names of roller coasters that need a service within 7 days' is tested using the sample data below.

```
Aftershock,Roller Coaster,510324,695,1.2m
Aqua Loop,Water,57288,542,0.9m
Asteroid Belt,Roller Coaster,551218,623,1.4m
```

Complete the trace table below to show the values up to the end of the third iteration.

(3 marks)

	If current category is 'Roller Coaster'	days	If (90 – days) is less than or equal to 7
1st iteration			
2nd iteration			
3rd iteration			

Candidate name_____ Candidate number_____

1e A new attraction is added to the theme park and the CSV file now has data for 27 attractions.

Evaluate the maintainability of your first sub-program 'Read data from file into parallel arrays', with reference to data structures and loops, based on this change.

(2 marks)

Candidate name_____ Candidate number_____

Task 2: database design and development (part A)

Gnome Sweet Gnome is a company that sells garden gnomes.

The owners want to create a relational database to store details of their customers, orders and garden gnomes. They have appointed a database developer to do this.

The developer asked employees about the features they would like in the completed database. The following is a summary of their responses.

I would like to view the details of the gnomes in order of popularity.

I want to see all orders including the date of the order.

I want to be able to produce a list of customers who have placed orders in a particular month.

I need to be able to find customer's details so that the company can contact them about special offers.

I would like to make changes to customer information.

I need to be able to view the total cost of each order.

I would like to give details of gnome names, descriptions and prices to the customers.

I would like to add new gnomes to the database as we get new stock.

2a A database needs to be created with four entities: Customer, CustOrder, GnomePurchase and Gnome.

Using the information gathered from the employees' comments, identify two additional functional requirements of the database.

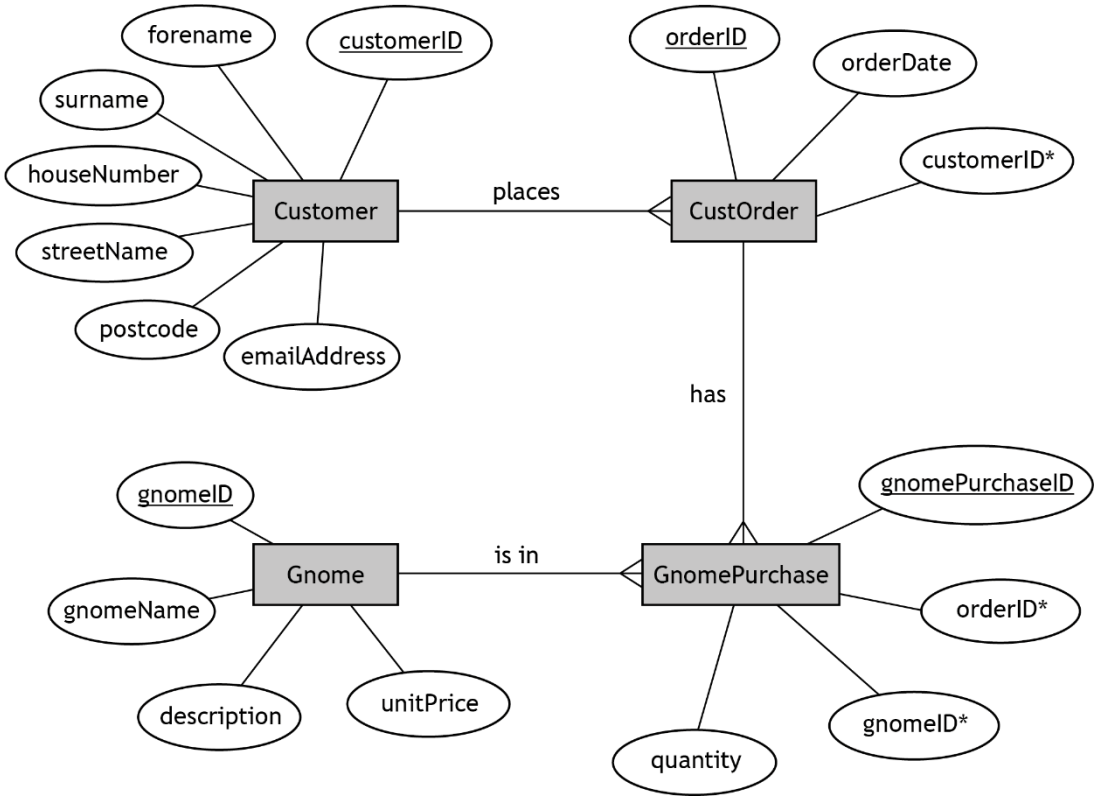
(2 marks)

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

Task 2: database design and development (part B)

Following further analysis, the entity-relationship diagram below is created.



This design is then implemented.

Your teacher or lecturer will provide you with a completed database.
This is a relational database with the following tables.

gnomeSales database			
Customer	**CustOrder**	**GnomePurchase**	**Gnome**
customerID	orderID	gnomePurchaseID	gnomeID
forename	orderDate	orderID*	gnomeName
surname	customerID*	gnomeID*	description
houseNumber		quantity	unitPrice
streetName			
postcode			
emailAddress			

2b Gnome Sweet Gnome would like to produce a list showing the number sold for each gnome with the word 'solar' in the description.

Implement the SQL statement to produce the following output.

(5 marks)

gnomeName	Total gnomes sold
Coimin	21
Maximilian	19
Danny	15

Print evidence of the implemented SQL statement and the output produced.

2c A discount voucher will be sent to customers who bought three or more of the most expensive gnome available, in a single order.

Implement the SQL statement(s) to produce the output shown below.

(4 marks)

emailAddress	orderID	Quantity
melina.santiago@coolmail.com	ord0061	5
maha.weber@yeehaa.com	ord0097	4

Print evidence of the implemented SQL statement(s) and the output produced.

2d A query is designed to add 20% VAT to all orders. The query is tested using order ord0024. The expected output is shown below.

Note: the database environment you use might output the total as 248.4

forename	surname	Total to Pay (£)
Sandy	Kerr	248.40

The SQL statement shown below is implemented.

```
SELECT forename, surname, (quantity * unitPrice * 1.2) AS [Total to Pay £]
FROM Customer, Gnome, GnomePurchase, CustOrder
WHERE CustOrder.orderID="ord0024"
AND Customer.customerID=CustOrder.customerID
AND CustOrder.orderID=GnomePurchase.orderID
AND Gnome.gnomeID=GnomePurchase.gnomeID;
```

The query to test the above SQL statement is provided with the database. When run, the actual output does not match the expected output.

Amend the query to produce the expected output as shown above.

(2 marks)

Print evidence of the amended SQL query and the output produced.

2e A customer requests a copy of a previous order for an insurance claim.

Evaluate the accuracy of output when running a new query to produce a copy of the original order by:

(i) explaining why the copy of the order may not reflect the price paid at the time

(1 mark)

(ii) describing how the database could be amended to rectify this

(1 mark)

Candidate name_____ Candidate number_____

Task 3: web design and development (part A)

Planet Games is a company that reviews computer games and reports on the latest gaming news. Their existing website is outdated and they want to create a new one.

The new website should:

- allow users to leave their own reviews
- keep reviews for a game hidden until selected (to avoid spoilers)

3a Using the information above, create two functional requirements for the website.

(2 marks)

- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

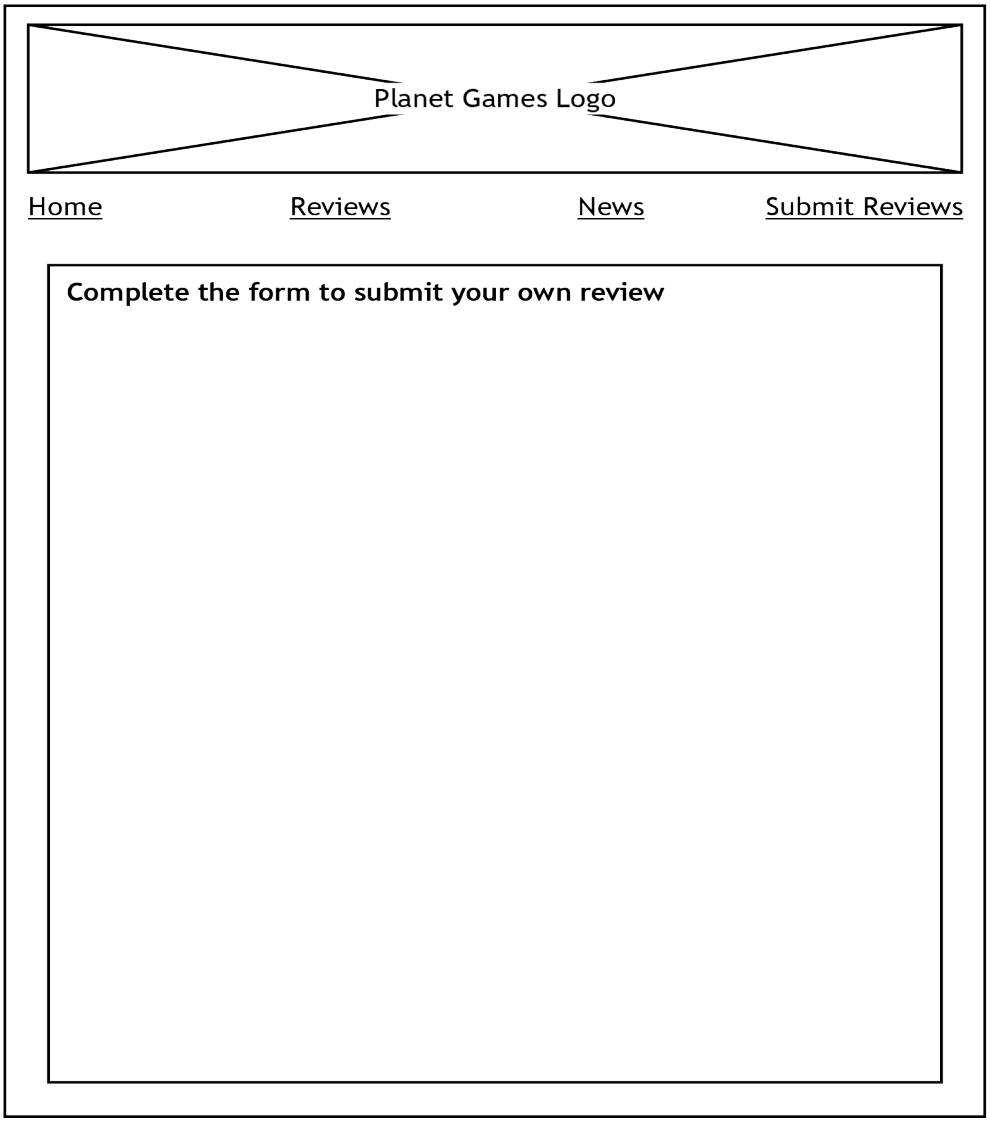
Task 3: web design and development (part B)

3b A form with the following information is required to allow users to leave their own reviews.

- The user's first name, surname and age range (16-18, 19-24, 25-29, 30+).
- The name of the game being reviewed.
- A rating from 1 to 10.
- A written review of the game.

Complete the wireframe below to show how the form could be designed.
The design should show validation.

(2 marks)



- Check your answers carefully, as you cannot return to part B after you hand it in.
- When you are ready, hand part B to your teacher or lecturer and collect part C.

Candidate name_____ Candidate number_____

Task 3: web design and development (part C)

Your teacher or lecturer will provide you with the complete website for Planet Games.

Open the 'home.html' page in a browser. Examine the home page and each of the pages on the website.

3c The 'Reviews' page currently has all the reviews shown on the screen at once.

Edit the code so that all reviews are hidden when the page is loaded and each review only appears when the corresponding image is clicked. Only one review should appear on the screen at a time.

(3 marks)

Print evidence of the:

- edited 'reviews.html' file
- 'Reviews' page, as viewed in a browser

3d The final design for the form needs to be added to the 'submitReview.html' page and is shown below:

Complete the form to submit your own review

First Name*
Text input box (max size 15 characters)

Surname*
Text input box (max size 15 characters)

Age*
4 Radio Buttons:

16-18, 19-24, 25-29, 30+

Genre Name*
Text input box (max size 20 characters)

Rating*
Number input box (1-10)

Your detailed review*
Text box area 4 rows 50 columns

Submit

Edit the 'submitReview.html' and the 'styles.css' files to implement the form, ensuring all validation is included and the layout matches the above design. All fields must be completed before the form is submitted.

(4 marks)

Print evidence of the:

- edited 'submitReview.html' and 'styles.css'files
- 'Submit Review' page, as viewed in a browser

3e Describe how you would fully test two **different types** of form elements on the 'Submit Review' page.

(2 marks)

Candidate name_____ Candidate number_____

3f Review the functional requirements of the website and evaluate if the website is fit for purpose.

(2 marks)

Functional requirements:

- The 'Home' page should contain information about Planet Games.
- The 'Submit Review' page should have a form to let users complete and submit their own reviews.
- The 'Reviews' page should contain:
  - a JavaScript function to reveal reviews when a corresponding image is clicked.
  - a form to allow users to comment on existing reviews.
- All pages should contain a navigation bar with links to all other pages.

Candidate name_____ Candidate number_____

Copyright acknowledgements

Task 1 – Tommy Alven/Shutterstock.com

Electronic files:

PremiumArt/Shutterstock.com
klyaksun/Shutterstock.com
Teguh Mujiono/Shutterstock.com
Kaleb-Silva/Shutterstock.com
Danilo Sanino/Shutterstock.com
Catalyst Labs/Shutterstock.com

Administrative information

Published: January 2023 (version 1.0)

History of changes

Version	Description of change	Date

Security and confidentiality

This document can be used by SQA approved centres for the assessment of National Courses and not for any other purpose.

This document may only be downloaded from SQA's designated secure website by authorised personnel.

© Scottish Qualifications Authority 2023