Candidate 1 evidence

Database design and development

Task 1: database design and development (part A)

Video bloggers (vloggers) create videos to upload to social media websites. Mirren promotes vloggers across Scotland. She keeps a record of vloggers and the details of their videos. Mirren names each video and rates them on a scale of 1 to 5 (one being the worst and five being the best). Videos may be up to 300 seconds in length.

Mirren decides to store these details in a database. The completed analysis of inputs is shown below.

Vlogger details:	Video details:
vloggerID forename surname username expertise	videoID vloggerID videoName duration dateCreated content rating

1a Complete the data dictionary for the Video entity.

(5 marks)

Entity name: Video					
Attribute name	Key	Type	Size	Required	Validation
videoID	PK	number		Y	
vloggerID	FK	number		Y	existing vloggerID from Vlogger table
videoName		text	30	Y	
duration		time		Y	
dateCreated		date		Y	
content		text	40	Y	
rating		number		Y	existing video rating from video table

- 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.

Vlogger

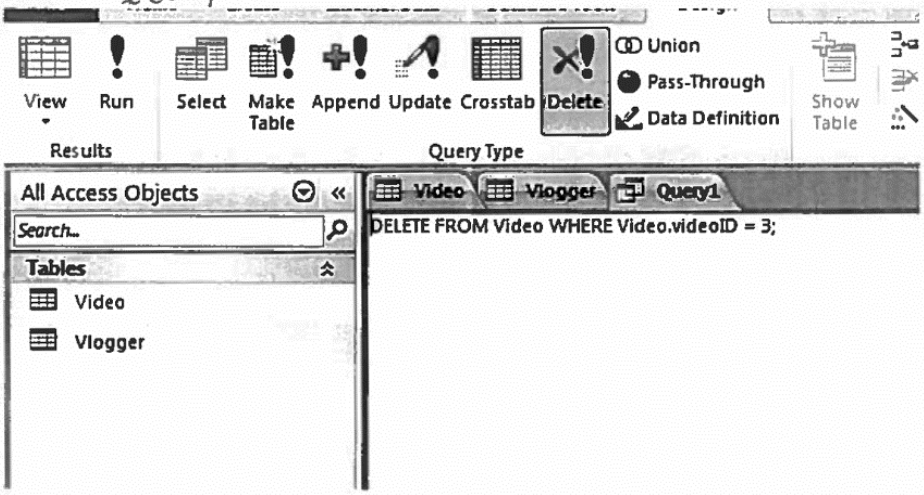
Field Name	Data Type	Description
vloggerID	Number	
forename	Text	
surname	Text	
username	Text	
expertise	Text	

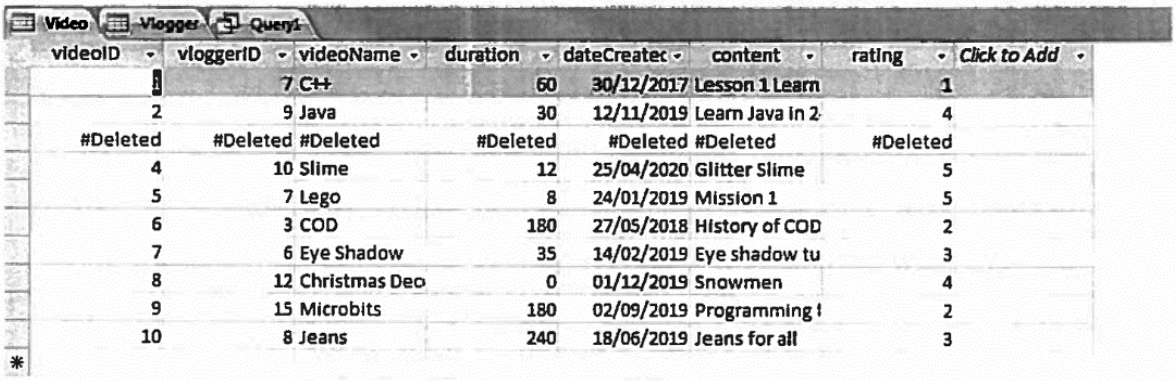
Field Properties

General | Lookup

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	>5 And <7
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

1cii)





Video | Vlogger | Query1

videoID	vloggerID	videoName	duration	dateCreatec	content	rating	Click to Add
1	7	C++	60	30/12/2017	Lesson 1 Learn	1	
2	9	Java	30	12/11/2019	Learn Java in 2	4	
#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	
4	10	Slime	12	25/04/2020	Glitter Slime	5	
5	7	Lego	8	24/01/2019	Mission 1	5	
6	3	COD	180	27/05/2018	History of COD	2	
7	6	Eye Shadow	35	14/02/2019	Eye shadow tu	3	
8	12	Christmas Deo	0	01/12/2019	Snowmen	4	
9	15	Microbits	180	02/09/2019	Programming t	2	
10	8	Jeans	240	18/06/2019	Jeans for all	3	

1(i)

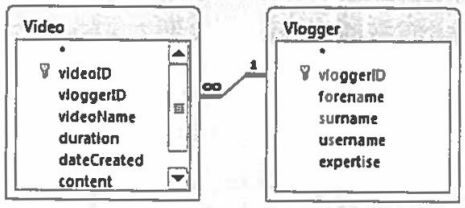
Video | Vlogger | Rating Query

```
SELECT Video.rating, Vlogger.username, Video.videoName
FROM Vlogger AND Video
WHERE Video.rating>3;
```

Video | Vlogger | Rating Query

rating	username	videoName
	Casing	Java
5	Joeing	Slime
5	Loging	Lego
4	Alding	Christmas Dec

Video | Vlogger | Rating Query



Field:	rating	username	videoName
Table:	Video	Vlogger	Video
Sort:			
Show:	☑	☑	☑
Criteria:	>3		
or:			

Vlogger

Field Name	Data Type	Description
vloggerID	Number	
forename	Text	
surname	Text	
username	Text	
expertise	Text	

Field Properties

General | Lookup

Field Size	15
Format	
Input Mask	AAAAAAAAAAAAAAA
Caption	
Default Value	
Validation Rule	"Programming" Or "Gaming" Or "Baking" Or "Crafts" Or "Makeup" Or "Clothes"
Validation Text	
Required	Yes
Allow Zero Length	Yes
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

Software design and development

2a Complete the table by filling in the missing input, process and output.

(3 Marks)

Input	
1.	Ask for amount of generated usernames
2.	Enter the first 3 letters of the student name
Process	
1.	Check length of partial student name
2.	~~Input validation - is student name 3 letters long?~~ Generate random ending
3.	Add the partial student name with the randomly generated ending from the stored list
Output	
1.	Print student name with generated ending

- 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.

task 2b.py

```
"""
Name: Generating Usernames
Date:26/2/19

Description: Assignment task 2b
"""

#initialise variables
ending= [""]*5
ending[0]= "ing"
ending[1]= "end"
ending[2]= "axe"
ending[3]= "gex"
ending[4]="goh"
noofnames = 0
endstudentname=""

#ask user for number of student usernames to be generated
print("How many usernames are to be generated")
noofnames= int(input())

#initialise variables
studentname =[""]*(noofnames)
lenstudentname =[0]*(noofnames)
randomnumber=[0]*(noofnames)

#for loop- only repeat for how many student names are generated
for loop in range(0,(noofnames)):
    #input validation
    valid=False
    while valid ==False:
    #ask user for first 3 letters of the student name
        print("What are the first 3 letters from the name of student", loop+1)
        studentname[loop]= str(input())

        #calculate length of string input
        lenstudentname[loop] = len(studentname[loop])

        #if statement- is input 3 letters long
        if lenstudentname[loop] <3 or lenstudentname[loop] >3:
            print("input invalid")
            print("input must be 3 characters long")
        else:
            print("input valid")
            valid= True

    #get random number
    import random

    #set range
    randomnumber[loop]= random.randint(1,5)
```

task 2b.py

```
#associating random number with each final 3 letters
if randomnumber[loop]==1:
    endstudentname= ending[0]
    #concatenating strings
    studentname[loop]= (studentname[loop],endstudentname)

elif randomnumber[loop]==2:
    endstudentname= ending[1]
    #concatenating strings
    studentname[loop]= (studentname[loop],endstudentname)

elif randomnumber[loop]==3:
    endstudentname= ending[2]
    #concatenating strings
    studentname[loop]= (studentname[loop],endstudentname)

elif randomnumber[loop]==4:
    endstudentname= ending[3]
    #concatenating strings
    studentname[loop]= (studentname[loop],endstudentname)

elif randomnumber[loop]==5:
    endstudentname= ending[4]
    #concatenating strings
    studentname[loop]= (studentname[loop],endstudentname)

#display username
print("Your final username is", studentname[loop])
```

2b Using the program design and refinements, implement the program in a language of your choice. Ensure the program matches the pseudocode provided.

(15 marks)

Print evidence of your program code.

2c Your program should be tested to ensure it will only accept 3 characters.

Complete the test table below

(2 marks)

Type of test	User input	Expected result	Actual result
Normal	ben – 3 characters long	Input accepted	Printout of final output to show that input is accepted.
Exceptional	benjamin – 8 characters long	Error message displayed	Printout to show that an error message is generated.

2d Test your program using the following student names.

Chris
Christina
Christopher
Chrethe
Chrisoula
Christie

Provide evidence of the inputs and outputs to show that you have completed the test.

(1 mark)

Task 2C

Printout Evidence

Type of test	User input	Expected result	Actual result
Normal	Ben	Input accepted	Input Accepted
Exceptional	Benjamin	Error Message Displayed	Error Message Displayed

```
How many usernames are to be generated
2
What are the first 3 letters from the name of student 1
ben
input valid
Your final username is ('ben', 'goh')
What are the first 3 letters from the name of student 2
benjamin
input invalid
input must be 3 characters long
What are the first 3 letters from the name of student 2
ben
input valid
Your final username is ('ben', 'gex')
>>>
```

Task 2D

Printout Evidence

```
How many usernames are to be generated
6
What are the first 3 letters from the name of student 1
chris
input invalid
input must be 3 characters long
What are the first 3 letters from the name of student 1
christina
input invalid
input must be 3 characters long
What are the first 3 letters from the name of student 1
christopher
input invalid
input must be 3 characters long
What are the first 3 letters from the name of student 1
chrethe
input invalid
input must be 3 characters long
What are the first 3 letters from the name of student 1
chrisoula
input invalid
input must be 3 characters long
What are the first 3 letters from the name of student 1
christie
input invalid
input must be 3 characters long
What are the first 3 letters from the name of student 1
```

2e With reference to your code and testing, evaluate your own program by commenting on the following:

Efficient use of programming constructs in your code.

(1 mark)

The program is efficient because it uses arrays which take up less storage as less variables are used. It also uses loops to save time and storage as there is no need to repeat /rewrite same code.

Robustness of your completed program

(1 mark)

The program is robust because it uses input validation. Instead of crashing ~~or~~ stopping when an invalid number is entered, it* repeats the question until a valid number is entered.

*displays an error message and then

The readability of your code

(1 mark)

The code is readable because it uses clear comments explaining processes throughout; whitespace to separate processes and make the program ~~names~~ look less messy; and meaningful variable names to make it easier to understand what each one is for.

Evaluate the fitness for purpose of the solution

(1 mark)

The program is fit for purpose because it meets the functional and end-user requirements decided in the analysis stage. It follows the structure of the pseudocode provided. The program is fit for purpose because it has been tested and all test results were positive.

Web design and development

Task 3: web design and development

Too Good to Throw Away is a charity clothes shop. They would like a web page to encourage donations of clothes.

It will have a:

- heading with the title "Too Good to Throw Away!"
- graphic of clothes
- coloured section with a subheading entitled "What we need".
- numbered list (from 1 to 5) detailing the items the charity shop would like donated
- coloured section with a subheading titled "What we have in stock"
- video showing the current stock.

3a State **two** functional requirements for this web page.

Functional requirement 1	(1 mark)
the website ~~play~~ will play a video	

Functional requirement 2	(1 mark)
Must have a ~~ordered~~ numbered list	

3c Describe two tests that could be performed on this web page.

Test 1

(1 mark)

making sure the vidios work

Test 2

(1 mark)

~~make sure~~ check it meets the require-ments.

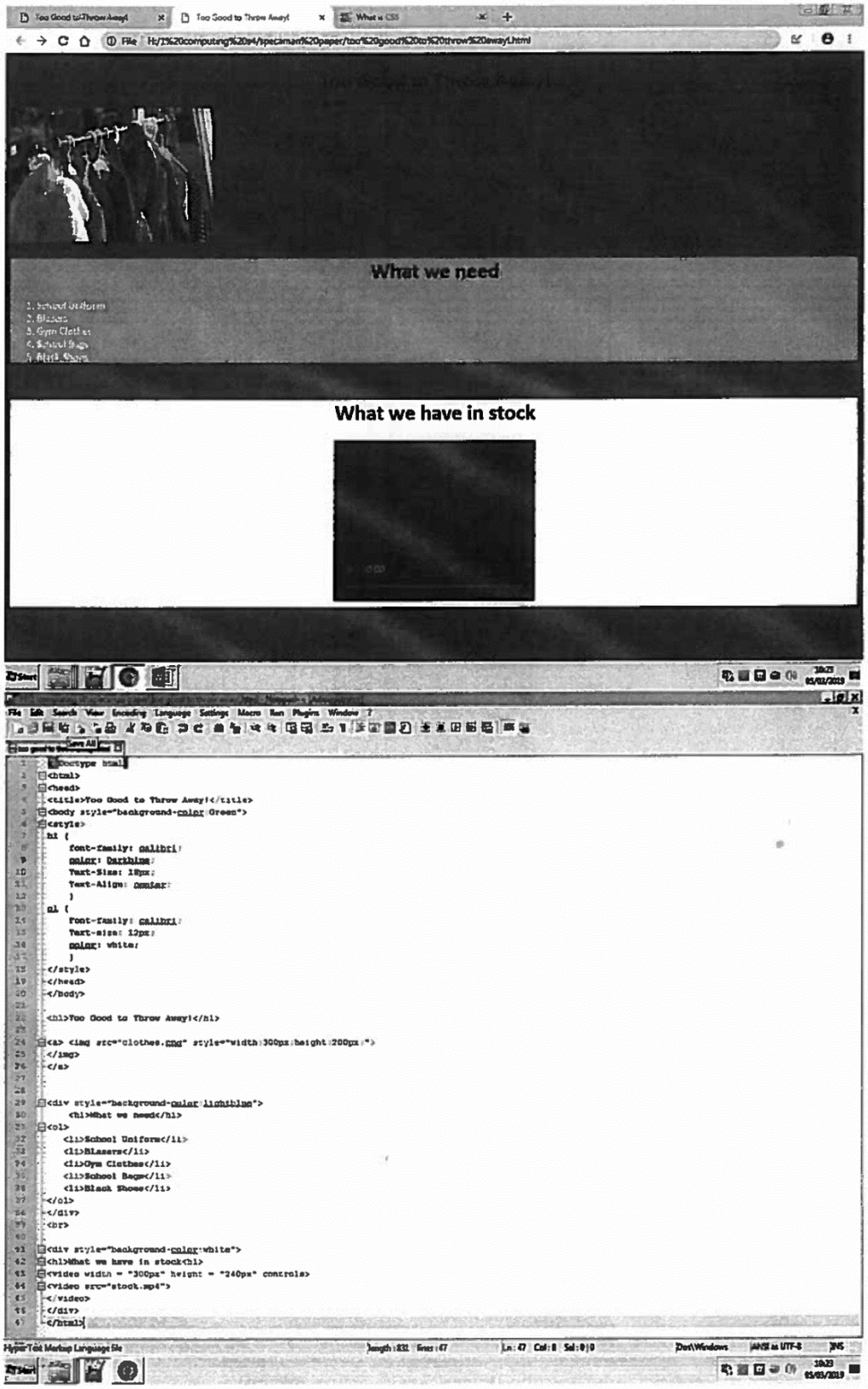
3d With reference to your solution, evaluate your web page by commenting on the following:

Fitness for purpose

(1 mark)

shows all the relavent information
matches the wire frame

3b


Too Good to Throw Away!
Too Good to Throw Away!
What is CSS
What we need
What we have in stock
<!Doctype html>
<html>
<head>
<title>Too Good to Throw Away!</title>
<body style="background-color:Green">
<style>
h1 {
font-family: calibri;
color: Darkblue;
Text-Size: 18px;
Text-Align: center;
}
ul {
font-family: calibri;
Text-size: 12px;
color: white;
}
</style>
</head>
</body>
<h1>Too Good to Throw Away!</h1>
<a> <img src="clothes.png" style="width:300px;height:200px;">
</img>

<h1>What we need</h1>

School Uniform
Blazers
Gym Clothes
School Bags
Black Shoes

<h1>What we have in stock</h1>
<video width = "300px" height = "240px" controls>
<video src="stock.mp4">
</video>

</html>