

# T LEVEL

*Technical Qualification in  
Digital Production, Design  
and Development*

## Student Exemplar Response Employer Set Project

Grade A

Version 1

© everything possible / Shutterstock,  
© Indypendenz / Shutterstock,  
© 3dreams / Shutterstock



## Contents

INTRODUCTION	2
QUESTION PAPER	3
MARK SCHEME	45
LEARNER WORK	55
MARKING GRID	94

## Introduction

This document contains a Student Exemplar Response portfolio of work and supporting documents for the Employer Set Project (ESP) for the T Level Technical Qualification. Included alongside the student portfolio submission are the question paper, mark scheme and mark grid.

The purpose of this material is to support providers and students in their understanding of the requirements of the ESP component and to prepare for future submissions. Providers may wish to use this publication to inform their teaching to cater for the ability range of candidates in their classes and the importance of teaching the practical skills and underpinning knowledge to ensure success in the ESP.

The student work submitted was marked and awarded at the grade documented in the mark grid for this document. It should be noted that the portfolio of work provided may include tasks (and elements of) which perform above, at and below the final awarded grade and the holistic grade for the portfolio is made up of the performance across all tasks. It is recommended that providers expose students to a wide range of scenarios that provide coverage of the outline content, so that students are well prepared for future series.

The portfolio provided is typical of a learner performing at the grade awarded for the Employer Set Project and should therefore indicate an indicative level of performance to achieve that grade.

*The exemplar student responses for the Employer Set Project have been taken from the 2022 summer assessment series, during which Ofqual asked awarding organisations' awarding committees to award more generously given the context of the pandemic and because these are new qualifications.*

**T Level Technical Qualification in Digital, Digital Production, Design  
and Development (Level 3)**

**May 2022**

**Controlled hours** N/A

**Paper  
reference**

**19538**

**Core: Employer Set Project**

**Pre-task: Familiarisation with the industry context**

**You do not need any other materials.**

### Information

- This booklet contains material for the completion of the set task under supervised conditions.
- This booklet is specific to each series and this material must only be issued to students who have been entered to undertake the task in the relevant series.
- This booklet should be kept securely until the start of the 16/05/2022 supervised assessment period.

*Turn over* ►

**W73767A**

©2022 Pearson Education Ltd.  
1/1/1/1



  
**Pearson**

## **Instructions to students**

The Familiarisation Set Task Brief will be issued one week before the start of Task 1 to allow you to familiarise yourself with the industry context of the business.

You should complete your familiarisation during the window of 16/05/2022 to 20/05/2022.

During the pre-release window you should spend time becoming familiar with the ways in which digital tools and technologies are used within the identified sector.

You are advised to spend around six hours on this task but may choose to spend more time if you wish.

Your centre will schedule a minimum of six hours to allow you to complete your familiarisation.

You are permitted to:

- access the internet
- work with others
- share and receive findings from the investigation you will be carrying out
- complete your investigation both during and outside of supervised sessions
- make notes about your findings.

You will not be allowed to take your notes into the supervised assessment sessions.

## Set Task Brief

You work for a software development company that has been offered the contract to develop the new digital solution for Newhaven Property Investments. Your manager has asked you to work as part of a group of developers to investigate the use of digital technology in the property sector in preparation for the start of the project and report back to your colleagues.

Newhaven Property Investments provides two main services:

- estate agency services for buying and selling properties
- advice on investment in real estate.

The company has offices in different regions around the country. Each region of the organisation currently operates independently, and with different approaches to forecasting and recording.

After reviewing its current systems, Newhaven Property Investments has decided that it needs to develop a new centralised digital solution.

The aims of the solution are to allow:

- customers and company representatives to communicate more effectively
- customers to:
  - be able to receive tailored advice on property sales and investments, based on region and budget
- management to:
  - access tools for carrying out business data analysis
  - help improve service around client needs (data analysis).

The requirements of the system are:

- a user interface (GUI) with appropriate accessibility features
- to provide customers with access to up-to-date information on sales and investments
- to allow staff to access information on sales and client investments
- to allow staff to interact with a dashboard of business information.

## **Activity**

Working in teams of four to six people, you must familiarise yourself with the use of digital solutions in the property sector.

To do this you may:

- use the information provided in the Set Task Brief
- carry out an independent investigation
- share and receive findings with others.

You may wish to consider:

- data analysis and modelling
- data and information security
- technical support solutions.

## **Outcomes for submission**

The purpose of this pre-release is to enable you to become familiar with the ways in which digital technology is used within the identified industry.

You are not required to submit any evidence for this pre-release.

You are permitted to make notes during the pre-release period but will not be allowed to take these into the assessment.



**T Level Technical Qualification in Digital, Digital Production, Design and Development (Level 3)**

**23 May 2022**

**Controlled hours** 3 hours

**Paper  
reference**

**19538**

**Core: Employer Set Project**  
**Task 1: Planning a project**

**You do not need any other materials.**

### Information

- This booklet contains material for the completion of the set task under supervised conditions.
- This booklet is specific to each series and this material must only be issued to students who have been entered to undertake the task in the relevant series.
- This booklet must be kept securely until the start of the timetabled assessment.

Turn over ►

**W73768A**

©2022 Pearson Education Ltd.  
1/1/1/1



W 7 3 7 6 8 A



**Pearson**



## Instructions to students

You must complete **ALL** parts of the activity within the assessment.

The task must be undertaken at the time and on the date specified by Pearson.

You will be given three hours for producing the outcomes for this task.

Your centre will advise you of when any supervised breaks have been scheduled.

The task must be completed under supervised conditions.

You are **not** permitted access to the internet during this task.

You are permitted to use **offline** versions of relevant software to produce evidence for this task.

Your work and any material provided must be kept securely at all times.

## Set Task Brief

You work for a software development company that has been offered the contract to develop the new digital solution for Newhaven Property Investments.

Your manager has asked you to analyse the potential costs, benefits and risks associated with the project and then produce a plan for the development of the new digital solution.

Your manager intends to use your plan during the next monthly meeting with the client.

## Activity

Using the Set Task Information provided your manager has asked you to produce:

- a Gantt chart to demonstrate how you would organise the development of the new digital solution
- a plan for the resources selected and associated costs
- a rationale that explains your planning approach and justifies the decisions you made.

(19)

## Outcomes for submission

Save your Gantt chart as a PDF file **and** as either:

- an Excel compatible spreadsheet file
- or MS Project compatible file (as appropriate).

Save your resource and cost plan as a PDF file **and** as:

- an Excel compatible spreadsheet file
- a MS Project compatible file (as appropriate).

Save your rationale as a PDF file.

All files should be saved in your folder for submission.

Use these naming conventions:

Task1GanttChart\_[Registration number #]\_[surname]\_[first letter of first name]

Task1ResourceCostPlan\_[Registration number #]\_[surname]\_[first letter of first name]

Task1Rationale\_[Registration number #]\_[surname]\_[first letter of first name]

## Set Task Information

You work for a software development company that has been offered the contract to develop the new digital solution for Newhaven Property Investments. Your manager has asked you to work as part of a group of developers to investigate the use of digital technology in the property sector in preparation for the start of the project and report back to your colleagues.

Newhaven Property Investments provides two main services:

- estate agency services for buying and selling properties
- advice on investment in real estate.

The company has offices in different regions around the country. Each region of the organisation currently operates independently, and with different approaches to forecasting and recording.

After reviewing its current systems, Newhaven Property Investments has decided that it needs to develop a new centralised digital solution.

The aims of the solution are to allow:

- customers and company representatives to communicate more effectively
- customers to:
  - be able to receive tailored advice on property sales and investments, based on region and budget
- management to:
  - access tools for carrying out business data analysis
  - help improve service around client needs (data analysis).

The requirements of the system are:

- a user interface (GUI) with appropriate accessibility features
- to provide customers with access to up-to-date information on sales and investments
- to allow staff to access information on sales and client investments
- to allow staff to interact with a dashboard of business information.

## Planning information

Newhaven Property Investments estimates that the new solution will allow an increase in revenue of 3% in its first year, 6% in its second year and an increase of 9% in its third year.

Newhaven Property Investments and your manager have provided you with information to help you produce your plan.

Company Financial Data	
Description	Amount (£)
Current annual revenue	912 000
Current outgoing costs	855 000
Forecast for development	Amount (£)
Purchasing of hardware	12 000
Training for staff	3 500
Software development company base fee	52 000
Software licensing (annual)	Option 1: Single fee 1 500 with three years' technical support  Option 2: Annual fee 250 without technical support
Upgrades to company infrastructure	5 000
Server upgrade	2 700
Your pay (project lead)	140 per day
Ongoing costs	Amount (£)
Maintaining servers and equipment (annual)	10 000

Due to current staff commitments, your manager has given you permission to make use of an outsourced team in order to complete the project.

Your project development team will consist of these new, temporary staff:

- one Senior Software Engineer
  - Weronika Marzanna
- two Junior Software Engineers
  - Terrence Bull
  - Nasira Amina
- one Hardware and Networking Technician
  - Susanne Morris
- one Database Engineer
  - Ayaru Thapa.

#### Profiles for the new staff

<b>Name</b>	Weronika Marzanna
<b>Title</b>	Senior Software Engineer
<b>Core skills</b>	<ul style="list-style-type: none"><li>• Extensive experience in deployment of software.</li><li>• Able to develop modules around data analysis and sales to a consistently high standard.</li><li>• Able to programme in multiple languages.</li></ul>
<b>Previous experience</b>	<ul style="list-style-type: none"><li>• Has worked as a software developer across a number of large-scale projects.</li><li>• Has some hardware testing experience from Level 3 studies, but this experience was several years ago.</li></ul>
<b>Cost</b>	£100 per hour

<b>Name</b>	Terrence Bull
<b>Title</b>	Junior Software Engineer
<b>Core skills</b>	<ul style="list-style-type: none"> <li>• Limited experience in deployment of software.</li> <li>• Has certification in multiple methods of software testing.</li> <li>• Can code in Python and C.</li> <li>• Has some experience of HTML, CSS and Javascript, and limited experience of backend database development.</li> </ul>
<b>Previous experience</b>	<ul style="list-style-type: none"> <li>• Used to work in a staff development training role.</li> <li>• Has worked as a user interface designer for a web development company.</li> </ul>
<b>Cost</b>	£45 per hour

<b>Name</b>	Nasira Amina
<b>Title</b>	Junior Software Engineer
<b>Core skills</b>	<ul style="list-style-type: none"> <li>• Has graduate level experience of programming and module development.</li> </ul>
<b>Previous experience</b>	<ul style="list-style-type: none"> <li>• Recent graduate.</li> <li>• Completed work experience for a real estate company.</li> </ul>
<b>Cost</b>	£35 per hour

<b>Name</b>	Susanne Morris
<b>Title</b>	Hardware and Networking Technician
<b>Core skills</b>	<ul style="list-style-type: none"> <li>• Has some intermediate experience of user interface development.</li> <li>• Has experience of developing test plans and hardware testing.</li> <li>• Has experience of updating and upgrading a range of infrastructure.</li> <li>• Developing sign off materials in line with a specific client's requirements.</li> </ul>
<b>Previous experience</b>	<ul style="list-style-type: none"> <li>• Worked as part of a team overseeing large-scale infrastructure upgrades.</li> </ul>
<b>Cost</b>	£70 per hour

<b>Name</b>	Ayaru Thapa
<b>Title</b>	Database Engineer
<b>Core skills</b>	<ul style="list-style-type: none"> <li>• Has certification in multiple methods of software testing.</li> <li>• Highly specialised in the development of database modules.</li> <li>• The ability to explain complicated processes in non-technical language.</li> </ul>
<b>Previous experience</b>	<ul style="list-style-type: none"> <li>• Has worked as a database engineer for 14 years.</li> </ul>
<b>Cost</b>	£75 per hour

The table shows the key tasks and the estimated number of work hours it will take to complete each task.

Task	Estimated hours
Upgrade infrastructure	30
Install and configure server	25
Develop module 1: Backend database	110
Develop module 2: Sales records	33
Develop module 3: User interface	50
Develop module 4: Data analysis	110
Develop module 5: Customer support	80
Deploy modules	6 (per module)
Installation of hardware	28
Testing of hardware	22
Unit testing	14 (per module)
Integration testing	15 (per module)
Fixing and regression testing for major fault	12 (per fault)
Fixing and regression testing for minor fault	5 (per fault)
Create a test plan	7
Staff training	30
User/acceptance testing	13
User training	15

It is assumed that:

- five minor faults will be found with each module
- a maximum of three major faults will be found throughout the whole project
- staff will work 7 hours a day 5 days a week.

Newhaven Property Investments has requested that the job is completed in 12 weeks.





**T Level Technical Qualification in Digital, Digital Production, Design and Development (Level 3)**

**26 May 2022**

**Controlled hours** 3 hours

**Paper  
reference**

**19538**

**Core: Employer Set Project**  
**Task 2: Identifying and fixing defects in  
an existing code**

**You must have:**

Task2\_Test\_Log\_Template.doc, Task2\_NonWorkingCode.txt.

### Information

- This booklet contains material for the completion of the set task under supervised conditions.
- This booklet is specific to each series and this material must only be issued to students who have been entered to undertake the task in the relevant series.
- This booklet must be kept securely until the start of the timetabled assessment.

Turn over ►

W73769A

©2022 Pearson Education Ltd.  
1/1/1/1



  
Pearson

**BLANK PAGE**

### Instructions for students

You must complete **ALL** parts of the activity within the assessment.

The task must be undertaken at the time and date specified by Pearson.

You will be given three hours for producing the outcomes for this task.

Your centre will advise you of when supervised breaks have been scheduled.

The task must be completed under supervised conditions.

You are **not** permitted access to the internet during this task.

You are permitted to use **offline** versions of relevant software to produce evidence for this task.

Files provided for use during this activity:

- Task2\_Test\_Log\_Template.doc
- Task2\_NonWorkingCode.txt

Your work and any material provided must be kept securely at all times.

## Set Task Brief

You are a member of the programming team that is developing a program to meet the requirements in the Set Task Information.

Your manager has asked you to look at some code that a Junior Software Developer has produced but is not yet functional. The code that is not yet functioning is provided for you in the *Task2\_NonWorkingCode.txt* file.

The code should meet the requirements in the Set Task Information.

## Activity

You will need to use:

- the information provided in the Set Task Information
- the non-functioning code provided in the file *Task2\_NonWorkingCode.txt*.

You must:

- produce and apply a test plan to identify the defects that are preventing the program code in the file *Task2\_NonWorkingCode.txt* from functioning
- apply a solution to fix the defects in the program code provided
- document the process that you followed to fix the code.

When applying a solution to fix the defects you must:

- ensure the code meets the requirements in the Set Task Information
- use Python 3 programming language
- follow accepted programming conventions
- test your solution to ensure that it functions as expected.

Two files are provided for use during this activity:

- Task2\_Test\_Log\_Template.doc
- Task2\_NonWorkingCode.txt

(21)

## Outcomes for submission

Save your code as PDF files **and** as .txt files.

Save your testing document as a PDF file.

All files should be saved in your folder for submission.

Use these naming conventions:

Task2code\_[doc #]\_[Registration number]\_[surname]\_[first letter of first name]

Task2\_Test\_Log\_[Registration number #]\_[surname]\_[first letter of first name]

## Set Task Information

Newhaven Property Investments has provided you with this information to develop your program.

## Requirements

You need to create a program that will be part of the new system's sales records module.

The program must allow the user to enter this information:

- sales information:
  - customer forename
  - customer surname
  - property to be purchased.

The program should then provide the user with a summary of the sale. The summary should show:

- customer forename
- customer surname
- price
- final total (with added fees).

The program should output the summary of the sale to the screen.

Values should be formatted as currency (GBP) to 2 decimal places.

The program should be able to handle user errors.

## Fees

The fees are calculated in this way:

- properties up to £100 000:
  - flat fee of 3%
- properties over £100 000:
  - fee of 3% of the first £100 000
  - plus 2% of the remaining value over £100 000.

```
houses = [['LONDON','Terraced', 3 735000], ['CARDIFF', 'Semi-Detached' 2, 100000], ['LEEDS','Terraced' 3,245000],
['LONDON','Semi-Detached' 1, 240000]]
sales = []
ourregions = ['LONDON', 'LEEDS', 'CARDIFF', 'BRISTOL']
property_types = ['TERRACED', 'SEMI-DETACHED','DETACHED']
```

```
def return_stock():
    print("CURRENT HOUSES FOR SALE \n\n REGION - HOUSE TYPE - BEDROOMS - COST")
    for i in houses:
        print (i)
```

```
def unique_regions():
    unique_list = []
    existing_regions = [item[0] for item in houses]
    for x in existing_regions:
        if x in unique_list:
            unique_list.append(x)
    print(unique_list)
```

```
def region_search():
    print("Available Regions")
    unique_regions()
    r_check = False

    while not r_check:
        region_select= input("Please enter region: ").capitalize()

        for x in houses:
            if region_select.upper() == x[0].upper():
                r_check = True
                if x[0] == region_select.upper():
                    print(x)

        if r_check == False:
            print("Entered region is not valid")
```

```
def show_sales():
    if len(sold) > 0:
        print("Forename Surname Property cost Total")
        for i in sales:
            print(i)
    else:
        print('no sales')
```

```
def house_sale():
    sale = []
    customer_forename = input('Please enter customer forename:')
    customer_surname = input('Please enter customer surname:')
```

```

for i, item in enumerate(houses, 1):
    print(i, item)

sel_check = False
while not sel_check:
    try:
        select = input('Please select a purchase')
        if select > 0 and select < len(houses):
            sel_check = True

    except:
        print('ERROR PLEASE ENTER A VALID PROPERTY')

sub_total = houses[select-1][3]
print(sub_total)
total_fees = 0

if sub_total > 100000:
    total_fees += 3000+(sub_total-100000) * 0.2
else:
    total_fees += sub_total *0.3

final_total = sub_total+total_fees
sale.append(customer_forename)
sale.append(customer_surname)
sale.append(sub_total)
sale.append(final_total)
sales.append(sale)

print('Customer Receipt\n\n FORENAME: {} SURNAME: {} PROPERTY COST: {} WITH STAMP DUTY:
{}'.format(*sales[-1]))
print('\nTRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES DATABASE\n')
print (houses[select])
del houses[select]

while True:
    menuselection = int(input(" WELCOME TO THE NEWHAVEN DASHBOARD \n\n Please select from the
following menu options \n\n"
        " 1: View current houses on market \n 2: Search for available houses in a region \n 3: Record"
        " a sale \n 4: Add a new property for sale \n 5:Show Sales \n 6: Exit"))

if menuselection == 1:
    return_stock
if menuselection == 2:
    region_search
if menuselection ==3:
    house_sale
if menuselection==4:
    show_sales

```





### Test 2 - Test Log

Description of test	Test data to be used (if required)	Expected outcome	Actual outcome	Comments and intended actions

Add more rows and tables as required



**T Level Technical Qualification in Digital, Digital Production, Design and Development (Level 3)**

**6 June 2022**

**Controlled hours** 3 hours

**Paper  
reference**

**19538**

**Core: Employer Set Project**  
**Task 3: Designing a solution**

**You must have:**

Task3\_data.csv

### Information

- This booklet contains material for the completion of the set task under supervised conditions.
- This booklet is specific to each series and this material must only be issued to students who have been entered to undertake the task in the relevant series.
- This booklet must be kept securely until the start of the timetabled assessment.

Turn over ►

W73770A

©2022 Pearson Education Ltd.  
1/1/1/1



  
**Pearson**

**BLANK PAGE**

### Instructions for students

You must complete the activity within the assessment.

The task must be undertaken at the time and on the date specified by Pearson.

You will be given three hours for producing the outcomes for this task.

Your centre will advise you of when any supervised breaks have been scheduled.

The task must be completed under supervised conditions.

You are **not** permitted access to the internet during this task.

You are provided with the file *Task3\_data.csv* for use during this task.

Your work and any material provided must be kept securely at all times.

## Set Task Brief

Part of the new digital solution requires your company to develop a data service that will turn raw data into meaningful information. The information will be used by the client to make data-driven decisions.

Your manager has decided to open the problem up to all developers to see if they can design a suitable solution.

Your manager has provided you with a set of requirements that he has agreed with the client.

## Activity

You must produce algorithm designs to show how you would implement the system and user requirements provided in the Set Task Information.

The algorithm designs should be of sufficient detail to:

- effectively communicate the intended solution
- allow the client to make informed decisions
- allow a third party to use the design documents to create the proposed solution.

You should use either pseudocode or flow charts to communicate your algorithms.

The file *Task3\_data.csv* contains the data that will be used by the solution. You should use this file to help you design your solution.

(17)

## Outcomes for submission

Save your completed design documents as PDF files.

All files should be saved in your folder for submission.

Use this naming convention:

Task3\_[document name]\_[Registration number #]\_[surname]\_[first letter of first name]

## Set Task Information

User requirements agreed with the client:

- The solution must allow users to select a region and show the trends for increases in property values over time.
- The solution must also:
  - be easy to use
  - display information in a meaningful way
  - make use of appropriate textual, numerical and graphical output in a way that would be relevant to the end user.

Note – the data shows the average percentage increase for all properties of each specific type for each region.

Region	CocRegion	Property T	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19	Aug-19	Sep-19	Oct-19	Nov-19	Dec-19	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20	Oct-20	Nov-20	Dec-20	Jan-21	Feb-21	Mar-21	Apr-21	May-21	Jun-21	Jul-21	Aug-21	Sep-21	Oct-21	Nov-21	Dec-21	Jan-22	Feb-22	Mar-22	Apr-22	May-22	
1	London	Bungalow	0.95	0.93	0.49	1.125	0.625	0.335	0.675	0.595	0.445	0.455	0.965	0.925	0.3	0.41	0.495	0.66	0.635	0.12	0.34	0.575	0.77	0.07	0.75	0.56	0.84	0.39	0.82	1.02	0.855	0.46	0.845	1.11	0.42	0.945	0.415	0.685	0.68	1.025	0.5	0.945	0.505	
1	London	Semi-Detac	1.075	0	0.515	0.615	0.15	0.66	0.09	0.185	0.46	0.935	0.7	0.885	0.83	0.455	0.495	0.415	0.6	0.325	0.56	0.115	0.56	0.8	0.685	0.435	0.415	0.5	0.29	0.4	0.3	0.59	0.695	0.575	0.765	0.445	0.875	0.845	0.635	0.57	0.12	0.97	1.01	0.23
1	London	Detached	1.02	0.49	0.905	0.87	1.435	0.585	0.99	0.915	1.23	0.57	0.68	0.335	1.25	0.765	0.55	0.79	0.835	0.995	0.57	0	1.115	0	1.195	1.13	0.74	1.05	0.84	1.005	0.315	0.475	0.905	0.79	0.425	0.455	0.75	1.47	0.265	0.77	1.12	0.835	1.495	
2	Bristol	Bungalow	0.745	0.615	0.42	0.475	0	0.705	0.375	0.77	0.79	0.47	0.35	0.465	0.57	0.71	0.575	0.43	0.345	0.425	0.19	0.115	0.41	0.565	0.61	0.455	0.67	0.525	0.505	0.515	0.775	0	0.43	0.485	0.005	0.75	0	0.725	0.625	0.665	0.265	0.345	0.495	
2	Bristol	Semi-Detac	0.37	0.505	0.74	0.4	0.48	0.47	0.27	0.695	0.63	0.68	0.535	0.42	0	0.515	0.525	0.615	0.405	0.18	0.92	0.73	0.365	0.455	0.435	0.59	0.22	0.49	0.495	0.48	0.44	0.68	0.38	0.83	0.615	0.145	0.545	0.27	0.71	0.165	0.395	0.86	0.725	
2	Bristol	Detached	0.605	0.555	1.175	0.635	0.445	0.5	0.275	0	0.61	0.525	0.14	0.61	0.675	0.175	0.965	0.35	0.625	0.11	0.78	0.62	0.255	0.44	0.18	1.235	0.995	0.955	0.9	0.335	0.66	0.675	0.83	1.115	0.875	0.3	0.245	0.47	0.925	0.58	0.67	0.445	0.995	
3	Cardiff	Bungalow	0.675	0.16	0.755	0.585	0	0.65	0.535	0.535	0.685	0.61	0.53	0.635	0.09	0.73	0.11	0.255	0	0.65	0.495	0.145	0.505	0.545	0.53	0.355	0.58	0.505	0.82	0.355	0.71	0.57	0.24	0.54	0.7	0.26	0.33	0.58	0.3	0.71	0.535			
3	Cardiff	Semi-Detac	0.465	0.435	0.415	0.3	0.395	0.235	0.875	0.685	0.395	0.24	0.605	0.47	0.665	0.495	0.48	0.235	0.575	0.675	0.81	0.35	0.555	0.605	0	0.015	0.36	0.7	0.395	0.36	0.57	0.24	0.605	0.585	0.555	0.65	0.39	0.38	0.83	0.49	0.68	0.495	0.57	
3	Cardiff	Detached	0.515	0.565	0.47	0.635	0.35	0.635	0.275	0.475	0.595	0.035	0.545	0.81	0.81	0.205	0.815	0.695	0.65	0.41	0.55	0.1	0.41	0.335	0.71	0.205	0.16	0.19	0.51	0.515	0.44	0.78	0.535	0.51	0.71	0.395	0.755	0.44	0.66	0.4	0	0.67	0.615	
4	Leeds	Bungalow	0.085	0.835	0.4	0.41	0.49	0.31	0.415	0.115	0.425	0.5	0.615	0.12	0.51	0.45	0.595	0.54	0.34	0.615	0.505	0.54	0.305	0.87	0.13	0.655	0.34	0.705	0.355	0.21	0	0.695	0.19	0.355	0.77	0.88	0.73	0.875	0.67	0.43	0.455	0.255	0.57	
4	Leeds	Semi-Detac	0.585	0.93	0.72	0.725	0.725	0.295	0.38	0.39	0.285	0.255	0	0.35	0.655	0.31	0.49	0.635	0.215	0.21	0.71	0.21	0.805	0.875	0.3	0.43	0.235	0.81	0.345	0.37	0.195	0.585	0.815	0.85	0.2	0.765	0	0.69	0.695	0.405	0.385	0.59	0.205	
4	Leeds	Detached	0.375	0	0.325	0.345	0.41	0.84	0.94	0.41	0.325	0.95	0.515	0.765	0.405	0.32	0.385	0.23	0.295	0.635	0.715	0.485	0.54	0.735	0.3	0.65	0.08	0.755	0.395	0.505	0.525	0.42	0.86	0.355	0.55	0.16	0.155	0.355	0.62	0.35	0.27	0.47	0.8	
5	Manchester	Bungalow	0.79	0.495	0.51	0.52	0.735	0.67	0.26	0.355	0.785	0.44	0.45	0.595	0.645	0.165	0.76	0.16	0	0.535	0.205	0.555	0.32	0.33	0.655	0.58	0.355	0.635	0.92	0.485	0.815	0.67	0.45	0.855	0.39	0.485	0.5	0.69	0.835	0.705	0.3	0.115	0.895	
5	Manchester	SemiDeta	0.38	0	0.72	0.58	0.74	0.225	0.555	0.445	0.725	0.665	0.93	0.83	0.59	0.685	0.38	0.22	0.225	0.625	0.145	0.735	0	0.64	0.465	0.41	0.085	0.635	0.465	0.61	0.175	0.46	0.8	0.875	0.435	0.35	0.725	0.49	0.195	0.28	0.325	0.5	0.37	
5	Manchester	Detached	0.69	0.29	0.24	0.495	0.475	0.7	0.49	0.475	0.32	0.415	0.3	0.615	0.585	0.695	0.125	0.205	0.315	0.465	0.09	0.64	0.285	0.365	0.91	0.53	0.51	0.52	0.185	0.51	0	0.515	0.465	0.375	0.26	0.575	0.455	0.765	0.495	0.515	0.63	0.63	0.195	
6	Birmingham	Bungalow	0.58	0.62	0.47	0.165	0.735	0.45	0.795	0.545	0.435	0.375	0.49	0.775	0.19	0.385	0.47	0.42	0.74	0.51	0.445	0.765	0.68	0.74	0.345	0.495	0.28	0.6	0.175	0.47	0.76	0.315	0.47	0.25	0.53	0.58	0.555	0.425	0.425	0.435	0.61	0.23	0.34	
6	Birmingham	Semi-Detac	0.62	0.315	0.765	0.88	0.775	0.45	0.79	0.26	0.395	0.685	0.92	0.29	0.575	0	0.385	0.585	0.795	0.68	0.36	0.485	0.54	0.43	0.775	0.555	0.5	0.43	0.55	0.295	0.81	0.37	0.645	0.49	0.435	0.835	0.445	0.565	0.33	0.355	0.505	0.765	0.515	
6	Birmingham	Detached	0.83	0.605	0.89	0.58	0	0.415	0.285	0.375	0	0.665	0.835	0.335	0.445	0.455	0.44	0.12	0.25	0.67	0.45	0.735	0.81	0.525	0.665	0.925	0.785	0.565	0.745	0.565	0.215	0.19	0.555	0.52	0.875	0	0.805	0.565	0.675	0.33	0.455	0.925	0.61	0.595
7	Edinburgh	Bungalow	0.71	0.32	0.46	0.17	0.66	0.695	0.235	0.22	0.665	0.41	0.125	0.71	0.785	0.75	0.5	0.925	0.755	0.355	0.98	0.575	0.835	0.58	0.655	0.715	0.54	0.705	0.255	0.555	0.745	0.76	0.51	0	0.495	0.235	0.385	0.655	0.625	0.465	0.32	0.11	0.785	
7	Edinburgh	Semi-Detac	0.57	0.22	0.395	0.465	0.585	0	0.455	0	0.31	0.785	0.785	0.855	0.525	0.45	0.31	0.775	0.63	0	0.64	0.65	0.165	0.15	0.32	0.765	0.255	0.06	0.14	0.45	0.545	0.23	0.62	0.335	0.445	0.68	0.68	0.425	0	0.14	0.755	0.795	0.705	
7	Edinburgh	Detached	0.39	0.665	0.55	0.41	0.815	0.195	0.77	0.715	0.33	0.77	0	0.515	0.56	0.23	0.715	0.465	0.385	0.69	0.495	0.34	0.835	0.785	0.86	0.575	0.475	0.695	0.6	0.16	0	0.645	0.93	0.855	0.71	0.63	0.41	0.555	0.43	0.485	0.23	0.29	0.36	
8	Glasgow	Bungalow	0.49	0.17	0.465	0.23	0.715	0.57	0.085	0	0.38	0.125	0.44	0.76	0	0.655	0.36	0.315	0.78	0.825	0.75	0.445	0.5	0.26	0.8	0.41	0.57	0.715	0.42	0.165	0.465	0.335	0.455	0.43	0.605	0.415	0.765	0.09	0.485	0.445	0.665	0.175	0.52	
8	Glasgow	Semi-Detac	0.735	0.62	0.505	0.31	0.44	0.97	0.715	0.43	0.575	0.71	0.46	0.735	0.7	0.37	0.195	0	0.835	0.525	0.385	0.585	0.7	0.48	0.605	0.355	0.62	0.235	0.855	0.58	0.605	0.555	0.155	0.54	0.235	0.895	0.435	0.605	0.595	0.41	0.645	0.535	0.835	
8	Glasgow	Detached	0.845	0.36	0.55	0.78	0.82	0.555	0.66	0.24	0.085	0.49	0.16	0.19	0.41	0.455	0.585	0.06	0.43	0.46	0.505	0.805	0.51	0.18	0.54	0.915	0.355	0.575	0.185	0.61	0.54	0.455	0.55	0.57	0.355	0.49	0.825	0.49	0.29	0.46	0.405	0.665	0.485	



**T Level Technical Qualification in Digital, Digital Production, Design and Development (Level 3)**

**10 June 2022**

**Controlled hours** 4 hours

**Paper  
reference**

**19538**

**Core: Employer Set Project**  
**Task 4a: Developing a solution**

**You must have:**

Task4a\_data.csv, Task4a\_Newhaven\_dashboard.py

### Information

- This booklet contains material for the completion of the set task under supervised conditions.
- This booklet is specific to each series and this material must only be issued to students who have been entered to undertake the task in the relevant series.
- This booklet must be kept securely until the start of the timetabled assessment.

Turn over ►

W73771A

©2022 Pearson Education Ltd.  
1/1/1/1/1/1



W 7 3 7 7 1 A



Pearson



**BLANK PAGE**

### Instructions for students

You must complete the activity within the assessment.

The task must be undertaken at the time and on the date specified by Pearson

You will be given four hours for producing the outcomes for this task.

Your centre will advise you of when any supervised breaks have been scheduled.

The task must be completed under supervised conditions.

You are **not** permitted access to the internet during this task.

You are provided with these files for use during this task:

- *Task4a\_data.csv*
- *Task4a\_Newhaven\_dashboard.py*

Your work and any material provided must be kept securely at all times.

## Set Task Brief

The design ideas for the solution that you worked on in Task 3 have been reviewed by your manager. A Senior Developer has coded the solution and your manager has presented it to Newhaven Property Investments.

Since then, Newhaven Property Investments has requested some additional functionality to be added to further develop the solution.

## Activity

The file *Task4a\_data.csv* contains the data that will be used by the solution.

The file *Task4a\_Newhaven\_dashboard.py* contains the code that the Senior Developer has produced.

Develop the programming code for the data service so that it meets the new system and user requirements provided in the Set Task Information.

During development of the solution make sure that you:

- consider the security requirements and use secure coding principles and practices to mitigate against potential threats and vulnerabilities
- ensure that your code is maintainable, readable and functional
- follow accepted programming conventions.

During this task you do **not** need to formally document your testing.

(34)

## Outcomes for submission

Save your code as PDFs and as .txt files.

Save a working version of your solution as a .py file.

All files should be saved in your folder for submission.

Use this naming convention:

- Task4acode [doc#]\_[Registration number]\_[surname]\_[first letter of first name]

You may use any appropriate folder structure and naming convention to save the working version of the solution.

### **Set Task Information**

Newhaven Property Investments has provided the new system requirements. These are:

- The existing code provided allow users to select a region and show the trends for increases in property values over time.
- The solution must also identify trends and patterns over time for:
  - specific property types of different sizes within a selected region
  - the region with the highest overall increase in property value.
- The solution must be secure.

Newhaven Property Investments has also provided user requirements for the solution. These are:

- be easy to use
- display information in a meaningful way
- make use of appropriate textual, numerical and graphical output in a way that would be relevant to the end user.

Note – the data shows the average percentage increase for all properties of each specific type for each region.

Region	Cod	Region	Property Ty	Rooms	Jan-19	Feb-19	Mar-19	Apr-19	May-19	Jun-19	Jul-19	Aug-19	Sep-19	Oct-19	Nov-19	Dec-19	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20	Oct-20	Nov-20	Dec-20	Jan-21	Feb-21	Mar-21	Apr-21	May-21	Jun-21	Jul-21	Aug-21	Sep-21	Oct-21	Nov-21	Dec-21	Jan-22	Feb-22	Mar-22	Apr-22	May-22	
1	London	Bungalow	2	0.79	0.92	0.93	1.09	0.64	0.2	0.27	0.67	0.05	0.38	0.81	0.66	0.01	0.8	0.37	0.15	0.89	0.24	0.06	0.24	0.89	0.03	0.26	0.33	0.92	0.18	1.29	0.87	1.29	0.31	0.5	1.13	0.1	1.11	0.59	0.31	1.04	0.96	0.55	1.1	0.71		
1	London	Bungalow	3	1.11	0.94	0.05	1.16	0.61	0.47	1.08	0.52	0.84	0.53	1.12	1.19	0.59	0.02	0.62	1.17	0.38	0	0.62	0.91	0.65	0.11	0.24	0.79	0.76	0.6	0.35	1.17	0.42	0.61	1.19	1.09	0.74	0.78	1.01	0.75	0.57	0.21	1.09	0.45	0.79	0.3	
1	London	Semi-Detac	2	1.12	0.08	0.67	0.71	0.06	0.64	0.03	0.32	0.66	0.93	1.03	1.06	0.57	0.19	0.47	0.23	0.64	0.56	0.3	0.18	0.94	0.42	0.51	0.81	0.3	0.17	0.76	0.11	0.04	0.72	0.55	0.71	0.79	1.01	0.75	0.57	0.21	1.07	0.92	1.06	0.14		
1	London	Semi-Detac	3	1.03	0.23	0.36	0.52	0.24	0.68	0.15	0.05	0.26	0.94	0.37	0.71	1.09	0.72	0.52	0.6	0.56	0.09	0.82	0.05	0.18	1.18	0.86	0.06	0.7	0.41	0.04	0.49	1.14	0.67	0.6	0.82	0.1	0.74	0.94	0.7	0.93	0.07	1.02	0.96	0.32		
1	London	Detached	2	1.22	0.27	0.41	0.88	1.31	1.03	0.64	0.15	0.87	0.6	1.04	0.74	1.15	0.26	0.95	0.8	0.39	0.75	0.33	0.26	1.12	1.33	1.59	0.71	0.01	0.1	1.41	0.68	0.3	0.95	1.51	1.47	0.3	1.48	0.69	1.55	0.27	0.33	1.53	1.33	1.28		
1	London	Detached	3	1.03	0.1	1.02	0.91	1.57	0.08	0.41	1.29	1.04	0.16	1.02	0.44	1.5	0.9	0.89	1.06	1.34	0.62	1.12	0.08	1.2	0.01	0.81	1.13	0.62	0.85	1.3	1.07	0.22	0.43	0.73	1.09	0.09	0.19	1.05	1.42	0.2	1.28	1.45	1.36	1.44		
1	London	Detached	4	1.01	0.88	0.79	0.83	1.3	1.09	1.57	0.54	1.42	0.98	0.34	0.23	1	0.63	0.21	0.52	0.33	1.37	0.02	0.39	1.03	0.85	1.58	1.13	0.86	1.25	0.38	0.94	0.41	0.52	1.08	0.49	0.76	0.72	0.45	1.52	0.33	0.26	0.79	0.31	1.55		
2	Bristol	Bungalow	2	0.5	0.64	0.17	0.41	0.78	0.75	0.1	0.79	0.66	0.92	0.55	0.81	0.48	0.79	1	0.02	0.18	0.21	0.11	0.12	0.04	0.42	0.51	0.88	0.42	0.91	0.19	0.18	0.64	0.89	0.35	0.38	0	0.81	0.04	0.5	0.56	0.73	0.25	0.65	0.65		
2	Bristol	Bungalow	3	0.99	0.59	0.67	0.54	0.43	0.66	0.65	0.75	0.92	0.02	0.15	0.12	0.66	0.63	0.15	0.84	0.51	0.64	0.27	0.11	0.78	0.71	0.71	0.03	0.92	0.14	0.82	0.85	0.91	0.09	0.51	0.59	0.01	0.69	0.4	0.95	0.69	0.6	0.28	0.04	0.34		
2	Bristol	Semi-Detac	2	0.57	0.42	0.93	0.65	0.07	0.87	0.14	0.6	0.42	0.82	0.51	0.17	0.27	0.82	0.53	0.45	0.17	0.12	0.99	0.5	0.59	0.64	0.78	0.91	0.1	0.04	0.32	0.27	0.19	0.77	0.19	0.73	0.32	0.05	0.8	0.15	0.74	0.12	0.07	0.95	0.49		
2	Bristol	Semi-Detac	3	0.17	0.59	0.55	0.15	0.89	0.07	0.4	0.79	0.84	0.54	0.56	0.67	0.81	0.21	0.52	0.78	0.64	0.24	0.85	0.96	0.14	0.27	0.09	0.27	0.34	0.94	0.67	0.69	0.59	0.57	0.93	0.91	0.24	0.29	0.39	0.68	0.21	0.72	0.77	0.96			
2	Bristol	Detached	2	0.11	0.85	1.29	1.14	0.27	0.13	0.55	0.31	0.14	0.61	0.28	0.45	1.23	0.7	1.25	0.43	0.57	0.65	1.16	0.61	0.95	0.92	1.17	0.34	0.37	0.08	0.23	0.86	1.13	1.27	0.95	1.11	0.17	0.85	0.02	1.07	0.56	0.08	0.23	0.41	0.46		
2	Bristol	Detached	3	1.06	0.67	1.06	0.36	0.17	0.4	0.43	0.82	0.55	0.02	0.19	0.62	0.72	0.18	0.86	0.09	1.12	0	1.11	0.09	0.44	0.57	0.25	1.2	0.73	0.93	1.28	0.3	0.26	0.77	0.58	1.29	0.76	0.45	0.39	0.1	1.06	0.7	1.2	0.59	1.01		
2	Bristol	Detached	4	0.15	0.44	1.29	0.91	0.72	0.6	0.12	0.26	0.67	1.03	0.09	0.6	0.63	0.17	1.07	0.61	0.13	0.22	0.45	1.15	0.07	0.31	0.11	1.27	1.26	0.98	0.52	0.37	1.06	0.58	1.08	0.94	0.99	0.15	0.1	0.84	0.79	0.46	0.14	0.3	0.98		
3	Cardiff	Bungalow	2	0.53	0.15	0.78	0.94	0.55	0.94	0.64	0.21	0.58	0.79	0.85	0.66	0.02	0.77	1.07	0.12	0.06	0.08	0.99	0.19	0.14	0.63	0.88	0.34	0.21	0.32	0.25	0.79	0.67	0.98	0.38	0.13	0.35	0.53	0.66	0.72	0.29	0.29	0.7	0.37	0.43	0.37	
3	Cardiff	Bungalow	3	0.82	0.17	0.73	0.23	0.77	0.36	0.43	0.86	0.79	0.43	0.21	0.61	0.16	0.69	0.1	0.45	0.92	0.31	0.8	0.15	0.38	0.21	0.72	0.5	0.84	0.76	0.85	0.04	0.44	0.76	0.35	0.19	0.15	0.42	0.68	0.23	0.37	0.46	0.23	0.99	0.7		
3	Cardiff	Semi-Detac	2	0.7	0.69	0.3	0.28	0.33	0.07	0.93	0.52	0.74	0.35	0.55	0.07	0.72	0.38	0.14	0.25	0.97	0.78	0.94	0.17	0.72	0.33	0.21	0.02	0.02	0.43	0.17	0.02	0.34	0.05	0.89	0.26	0.87	0.54	0.73	0.37	0.98	0.55	0.72	0.36	0.58		
3	Cardiff	Semi-Detac	3	0.23	0.18	0.53	0.32	0.46	0.4	0.82	0.85	0.05	0.13	0.66	0.87	0.61	0.61	0.82	0.22	0.18	0.57	0.68	0.53	0.39	0.88	0.56	0.01	0.7	0.97	0.62	0.7	0.8	0.43	0.32	0.91	0.24	0.76	0.05	0.39	0.68	0.43	0.64	0.63	0.56		
3	Cardiff	Detached	2	0.35	0.85	0.72	0.81	0.3	0.29	0.38	0.83	0.88	0.15	0.44	0.33	0.55	0.61	0.13	0.33	0.18	0.52	0.24	0.06	0.18	0.13	0.53	0.66	0.62	0.08	0.77	0.2	0.82	0.7	0.44	0.26	0.84	0.61	0.66	0.79	0.86	0.33	0.12	0.32	0.25		
3	Cardiff	Detached	3	0.44	0.36	0.37	0.79	0.7	0.83	0.4	0.39	0.46	0.07	0.67	0.93	0.83	0.33	0.76	0.92	0.94	0.68	0.29	0.17	0.72	0.58	0.99	0.12	0.27	0.35	0.76	0.45	0.19	0.76	0.19	0.73	0.7	0.27	0.55	0.15	0.65	0.02	0.42	0.9	0.84		
3	Cardiff	Detached	4	0.59	0.77	0.57	0.48	0	0.44	0.15	0.56	0.73	0	0.42	0.69	0.79	0.08	0.87	0.47	0.36	0.14	0.81	0.03	0.1	0.09	0.43	0.29	0.05	0.03	0.26	0.58	0.69	0.8	0.88	0.29	0.72	0.52	0.96	0.73	0.67	0.78	0.17	0.44	0.39		
4	Leeds	Bungalow	2	0.02	0.89	0.51	0.48	0.74	0.22	0.03	0.1	0.51	0.52	0.79	0.1	0.85	0.72	0.37	0.51	0.31	0.43	0.26	0.67	0.56	0.98	0.11	0.42	0.6	0.5	0.38	0.09	0.68	0.58	0.03	0.56	0.96	0.9	0.77	0.86	0.89	0.81	0.81	0.39	0.63		
4	Leeds	Bungalow	3	0.15	0.78	0.29	0.34	0.24	0.4	0.8	0.13	0.34	0.48	0.44	0.14	0.17	0.18	0.82	0.57	0.37	0.8	0.75	0.41	0.05	0.76	0.15	0.89	0.08	0.91	0.33	0.33	0.33	0.42	0.24	0.27	0.74	0.84	0.18	0.95	0.38	0.72	1	0.58	0.69	0.34	0.23
4	Leeds	Semi-Detac	2	0.98	0.88	0.75	0.86	0.69	0.13	0	0.03	0.3	0.32	0.11	0.6	0.91	0.02	0.05	0.33	0.13	0.33	0.74	0.4	0.84	0.88	0.47	0.24	0.32	0.86	0.35	0.42	0.24	0.27	0.74	0.84	0.18	0.95	0.38	0.72	1	0.58	0.69	0.34	0.23		
4	Leeds	Semi-Detac	3	0.19	0.98	0.69	0.59	0.76	0.46	0.76	0.75	0.27	0.19	0.82	0.1	0.4	0.6	0.93	0.94	0.3	0.09	0.68	0.02	0.77	0.87	0.13	0.62	0.15	0.76	0.34	0.32	0.15	0.9	0.89	0.86	0.22	0.58	0.81	0.66	0.39	0.23	0.08	0.84	0.18		
4	Leeds	Detached	2	0.64	0.11	0.14	0.48	0.71	0.66	0.3	0.87	0.29	0.62	0.7	0.31	0.95	0.46	0.31	0.36	0.49	0.31	0.63	0.2	0.09	0.61	0.09	0.26	0.52	0.28	0.42	0.27	0.43	0.02	0.92	0.07	0.58	0.09	0.48	0.84	0.58	0.4	0.3	0.46	0.98		
4	Leeds	Detached	3	0.09	0.58	0.23	0.52	0.75	0.98	0.92	0.01	0.19	0.93	0.1	0.72	0.08	0.34	0.12	0.24	0.21	0.45	0.98	0.12	0.6	0.78	0.57	0.33	0.06	0.87	0.4	0.63	0.67	0.38	0.98	0.31	0.19	0.22	0.09	0.24	0.7	0.01	0.44	0.41	0.92		
4	Leeds	Detached	4	0.66	0.52	0.42	0.17	0.07	0.7	0.96	0.81	0.46	0.97	0.93	0.81	0.73	0.3	0.65	0.22	0.38	0.82	0.45	0.85	0.48	0.69	0.03	0.97	0.1	0.64	0.39	0.38	0.38	0.46	0.74	0.4	0.91	0.1	0.22	0.47	0.54	0.69	0.1	0.53	0.68		
5	Mancheste	Bungalow	2	0.87	0.62	0.68	0.37	0.93	0.46	0.04	0.33	0.93	0.39	0.41	0.68	0.73	0.05	0.99	0.29	0.68	0.65	0.13	0.22	0.58	0.28	0.71	0.94	0.2	0.57	0.94	0.76	0.85	0.73	0.82	0.92	0.03	0.36	0.4	0.57	0.89	0.79	0.18	0.11	0.81		
5	Mancheste	Bungalow	3	0.71	0.37	0.34	0.67	0.54	0.88	0.48	0.38	0.64	0.49	0.49	0.51	0.56	0.28	0.53	0.03	0.95	0.42	0.28	0.89	0.06	0.38	0.6	0.22	0.51	0.7	0.9	0.21	0.78	0.61	0.08	0.79	0.75	0.61	0.6	0.81	0.78	0.62	0.42	0.12	0.98		

```

import datetime
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('Task_4a.csv')

def mainmenu():
    print("\t\t****Welcome to the Dashboard****")
    print('1) Return all current data')
    print('2) Return data for a specific region')
    return int(input(""))

def alldata():
    print(df)

def region_check(region, startdate, enddate): # region, startdate, enddate

    df1 = df.loc[:, startdate:enddate]
    df2 = df.loc[:, 'Region Code':'Rooms']

    result = pd.concat([df2, df1], axis=1, join='inner').where(df2["Region"] == region)
    result = pd.DataFrame(result)
    result.dropna(inplace=True)
    print(result)
    ave = df1.mean()
    ave.plot()
    plt.show()
    return result

x = mainmenu()
while x == 1 or x == 2:
    if x == 1:
        alldata()

    elif x == 2:
        while True:
            print()

            region = input("Please enter the name of the region you would like to check:")
            region = region.capitalize()
            if region in df.Region.values:
                while True:
                    startdate = input("PLEASE ENTER A START DATE AS MONTH-YEAR e.g. JAN-20")
                    startdate = startdate.capitalize()
                    if startdate not in df.columns:
                        print("Error start date not found")
                    else:
                        while True:
                            enddate = input("PLEASE ENTER AN END DATE AS MONTH-YEAR e.g. JAN-20")

```

```
    enddate = enddate.capitalize()
    if enddate not in df.columns:
        print("Error end date not found")
    else:
        region_check(region, startdate, enddate)
        break
    break
break
else:
    print("Region not found")

x = mainmenu()
```



**T Level Technical Qualification in Digital, Digital Production, Design and Development (Level 3)**

**13 June 2022**

**Controlled hours** 1 hour 30 minutes

**Paper  
reference**

**19538**

**Core: Employer Set Project**  
**Task 4b: Reflective evaluation**

**You must have:**

'read only' copy of the evidence you produced in Task 4a to assist you with your evaluation.

**Information**

- This booklet contains material for the completion of the set task under supervised conditions.
- This booklet is specific to each series and this material must only be issued to students who have been entered to undertake the task in the relevant series.
- This booklet must be kept securely until the start of the timetabled assessment.

*Turn over* ►

**W73772A**

©2022 Pearson Education Ltd.  
1/1/1/1/1



  
**Pearson**



**BLANK PAGE**

### Instructions for Students

You must complete the activity within the assessment.

The task must be undertaken at the time and on the date specified by Pearson.

You will be given one hour and 30 minutes for producing the outcomes for this task.

The task must be completed under supervised conditions.

You are **not** permitted access to the internet during this task.

Your centre will provide you with '**read only**' copy of the evidence you produced in **Task 4a** to assist you with your evaluation.

You are not allowed to make changes to evidence produced in previous tasks.

Your work and any material provided must be kept securely at all times.

## Set Task Brief

The project is now complete and is in the process of being handed over to Newhaven Property Investments.

Your company's policy insists that all employees reflect on the quality of their work at the end of every project.

Your manager has asked you to review the quality of your work.

Your manager has provided you with the system and user requirements to help you.

## Activity

Produce an evaluation of the solution you produced in **Task 4a**.

Your evaluation should justify:

- how well the solution you produced meets:
  - system requirements
  - user requirements
- how the solution could be further developed.

You have been provided with a copy of the system and user requirements in the Set Task Information to help you with your evaluation.

Your evaluation should be supported by specific examples from your work.

(9)

## Outcomes for submission

Save your report as a PDF file.

Your file should be saved in your folder for submission.

Use this naming convention:

Task4b\_Review\_[Registration number #]\_[surname]\_[first letter of first name]

## Set Task Information

The system requirements provided for **Task 4a** were:

- The existing code provided allows users to select a region and show the trends for increases in property values over time.
- The solution must also identify trends and patterns over time for:
  - specific property types of different sizes within a selected region
  - the region with the highest overall increase in property value.
- The solution must be secure.

The user requirements for the solution must:

- be easy to use
- display information in a meaningful way
- make use of appropriate textual, numerical and graphical output in a way that would be relevant to the end user.

Note – the data shows the average percentage increase for all properties of each specific type for each region.

# T Level Technical Qualification in Digital Production, Design and Development

## Mark Scheme (Results)

June 2022

Employer Set Project

## General Marking Guidance

- All learners must receive the same treatment. Examiners must mark the first learner in exactly the same way as they mark the last.
- Mark schemes should be applied positively. Learners must be rewarded for what they have shown they can do rather than penalised for omissions.
- Examiners should mark according to the mark scheme not according to their perception of where the grade boundaries may lie.
- All marks on the mark scheme should be used appropriately.
- All the marks on the mark scheme are designed to be awarded. Examiners should always award full marks if deserved. Examiners should also be prepared to award zero marks if the learner's response is not rewardable according to the mark scheme.
- Where judgement is required, a mark scheme will provide the principles by which marks will be awarded.
- When examiners are in doubt regarding the application of the mark scheme to a learner's response, a senior examiner should be consulted.
- Crossed out work should be marked **unless** the learner has replaced it with an alternative response.
- Accept incorrect/phonetic spelling (as long as the term is recognisable) unless instructed otherwise.

## Levels-Based Mark Scheme Guidance

Levels-based mark schemes (LBMS) have been designed to assess students' work holistically. They consist of two parts:

### 1) Indicative content

Indicative content reflects content-related points that a student might make but is not an exhaustive list. Nor is it a model answer. Students may make some or none of the points included in the indicative content as its purpose is as a guide for the relevance and expectation of the responses. Students must be credited for any appropriate response.

### 2) Levels-based descriptors

Each level is made up of a number of traits which when combined together articulate the quality of response that a student needs to demonstrate. The traits progress across the levels to demonstrate the different expectations of each level. When using a levels-based mark scheme, the 'best fit' approach should be used.

## Applying the levels-based descriptors

Examiners should take a 'best fit' approach to determining the mark.

- Examiners should first make a holistic judgement on which level most closely matches the student's response. Students will be placed in the level that best describes their answer. Answers can display characteristics from more than one level, and where this happens markers must use any additional guidance (e.g. weighting of traits) and their professional judgement to decide which level is most appropriate.
- The mark awarded within the level will be decided based on the quality of the answer and will be modified according to how securely all traits are displayed at that level:
  - Marks will be awarded at the top of that level if the student has evidenced each of the descriptor traits securely.
  - Where the response does not securely meet all traits, the marks should be awarded based on how closely the descriptor has been met.

## Task 1- Planning a project

### Indicative content and marker guidance

NOTES: COSTS – DON'T TAKE VALUES AS GIVEN – LOOK AT FORECASTS AND PROJECT COSTS

RATIONALE – SHOULD DISCUSS THEIR CHOSEN PLANNING METHODS – CAN BE IN DEPTH BUT

SHOULD REFLECT THEIR IMPLEMENTATION BE WARY OF WHICH SOFTWARE APPLICATION

YOU USE TO CHECK PROJECT FILES

Gantt Chart:

- Expect to see tasks broken down into smaller chunks where sensible for example:
  - May show use of an Agile approach (or similar)
  - Large modules (e.g. backend database, data analysis etc.) broken down into multiple sections of development and unit testing with logical resources being applied to tasks – look out for learners applying the same developer to test the modules – can be ok if justified through testing experience
  - When splitting tasks learners should show an understanding of how total development hours should be split between the team working on it.
  - A task related to the choice of software licensing should be seen here – single fee with technical support or annual fee without.
- There should be sensible use of concurrent and serial tasks for example:
  - Logical task sequences e.g. server setup/installed before module deployed.
  - Showing that as one unit is being tested, development could be taking place with other team members
  - Integration testing would be expected after specific modules have been unit tested
  - At the higher end, expect to see consideration around the testing time for each module and how this could be split up to allow testing along with other modules.
  - Possible to see an agile approach on a granular level e.g. per module as well as the project as a whole
  - Should show some awareness of the requirements of tasks having predecessor requirements
- There is no single correct way to organize the plan but task orders should be sensible for example 'create a test plan' should occur BEFORE testing commences, staff training would occur much later in the process when the system is nearer completion etc.
- The order and implementation of the project may vary significantly depending on the SDLC approach that they are taking (check against learner rationale). For example:
  - a RAD/agile that looks at a Minimum viable product (MVP) they may 'Deploy' some of the modules very early on, after a short portion of development time, and then test and develop further, deploying more modules as they go.
  - Or they may decide that their approach is to have most modules mostly developed and then deployed and tested later.

The rationale will show the reasoning for the chosen project development approach demonstrated in the Gantt chart. Points the learners may consider, although some of these will vary depending on the choices learners make in terms of organisation. These include but are not limited to:

- Staff (skills, industry expertise, experience)
  - The senior software engineer has worked on large scale projects and has experience of deployment, as well as specific experience of developing data analysis modules.
  - The junior software engineer (TB) has less experience but has some clear experience around staff development and UI development (though this is in a different industry), similarly, there is experience of working on a backend database here, but this is limited, which could mean carrying out work supervised.
  - The junior software engineer (NA) has limited applicable experience but is a cheaper option – if made use of, could see work being checked or overseen at intervals in the project by a more senior engineer.
  - The hardware and networking technician has the clear expertise in the field of overseeing large scale infrastructure upgrades so expect to see her in these tasks – if assigned elsewhere expect to see clear justifications in rationale – could be seen around the testing elements
  - The database engineer has lots of testing experience – as well as working on the database module, expect to see this team member testing across the range of modules. There is an option to use as part of staff training but would need to justify over the cheaper option (TB).
  - Testing should be allocated to members of staff in a sensible manner – in some cases, in learner work, testers are being allocated as they are the developer – in some cases this can make sense – if clearly justified in the rationale
- Resources
  - Learners should rationalize the choice of allocation of tasks to different members of staff (see above)
  - A justification of which licensing cost they opt for
  - Though costings vary – should show some logical methods used to calculate.

- Time scales and costs
    - Learners should identify if they made the deadline or not. The number of hours for the project and available time is very close – with consideration of predecessor tasks. The timeline is reasonable, expect some learners to come in under by suitably dividing tasks amongst appropriate staff.
    - More contingency time may be needed on jobs performed by the Junior staff members/senior members assigned to oversee work
    - Learners should discuss the cost of their solution with reference to how these costs were arrived at, and if the projected costs and increase in revenue make the project feasible – showing awareness that the project benefits will be seen over a number of years
    - Learners could look to justify using less experienced members of staff in order to keep costings down, though mitigation (e.g. overview by the senior, more time to allow for development) and rationale should be considered
  - Risks
    - Some staff have key skills but from different industries
    - It is unrealistic to assume that there will be no staff absence in what will be a large-scale project - wise to build contingency time into the project as a whole and/or the tasks assigned to each member of the team – this would be for higher level learners
    - Time scale consideration around getting the project complete on time
- In some cases, learners will write a lot for the rationale, but this may be almost a description of what they did, or a transfer of staff profiles – try to focus on aspects of the learner work that shows depth of consideration of their plans

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1-2	3-4	5-6
Gantt chart	No rewardable material	Project tasks are somewhat organised in logical and efficient manner making some use of an appropriate SDLC model to provide some accurate prediction of the project’s timescales  Resources have sometimes been assigned to project tasks effectively but there are some major and/or significant errors or omissions	Project tasks are organised in a mostly logical and efficient manner making use of an appropriate SDLC model to provide mostly accurate predictions of the project’s timescales  Resources have mostly been assigned to the project tasks effectively, but there are some minor errors or omissions.	Project tasks are organised in a thoroughly logical and efficient manner using an appropriate SDLC model to provide thoroughly accurate predictions of the project’s timescales  Resources have consistently been assigned to the project tasks effectively.
	0	1	2-3	4
Resource and cost plan		Some correct resources and accurate costs have been added to the plan resulting in an estimate of limited accuracy.	Mostly correct resources and accurate costs have been added to the plan resulting in an estimate that is largely accurate.	Fully correct resources and accurate costs have been added to the plan resulting in an accurate estimate.
	0	1-3	4-6	7-9
Rationale	No rewardable material	Rationale for project planning decisions demonstrate some effective consideration of: <ul style="list-style-type: none"> <li>• cost, risks and benefits</li> <li>• order and timing of tasks</li> <li>• selection and allocation of resources</li> <li>• dependencies and prerequisites.</li> </ul>	Rationale for project planning decisions demonstrate mostly effective consideration of: <ul style="list-style-type: none"> <li>• cost, risks and benefits</li> <li>• order and timing of tasks</li> <li>• selection and allocation of resources</li> <li>• dependencies and prerequisites.</li> </ul>	Rationale for project planning decisions demonstrate a thorough and perceptive consideration of: <ul style="list-style-type: none"> <li>• cost, risk and benefits</li> <li>• order and timing of tasks</li> <li>• selection and allocation of resources</li> <li>• dependencies and prerequisites.</li> </ul>



Task 2- Identifying and fixing defects in existing code

Indicative content and marker guidance		
Line number	Description of error	Possible fix
	houses = [['LONDON', 'Terraced', 3 735000], Missing, after 3	
3	ourregions = ['LONDON', 'LEEDS', 'CARDIFF', 'BRISTOL'] Array not used	MAY CHOOSE TO REMOVE, MAY CHOOSE TO INCORPORATE
	<p style="text-align: center;">before</p> <pre>select = input('Please select a purchase') if select &gt; 0 and select &lt; len(houses):     sel_check = True</pre> <p>Extra )</p>	Removal of extra bracket - NOTE HOWEVER THIS IS NEEDED FOR THE FIX FOR AN INT INPUT
14	<pre>def unique_regions():     unique_list = []     existing_regions = [item[0] for item in houses]     for x in existing_regions:         if x in unique_list:             unique_list.append(x)     print(unique_list)</pre> <p>Should be not in</p>	<pre>def unique_regions():     unique_list = []     existing_regions = [item[0] for item in houses]     for x in existing_regions:         if x not in unique_list:             unique_list.append(x)     print(unique_list)</pre>
	<pre>def show_sales():     if len(sold) &gt; 0:         print("Forename Surname Property cost Total")         for i in sales:             print(i)     else:         print('no sales')</pre> <p>Should be checking length of sales not sold</p>	<pre>def show_sales():     if len(sales) &gt; 0:         print("Forename Surname Property cost Total")         for i in sales:             print(i)     else:         print('no sales')</pre>
47	Should check as int input	<pre>try:     select = int(input('Please select a purchase'</pre>

<pre>sel_check = False while not sel_check:     try:         select = input('Please select a purchase')         if select &gt; 0 and select &lt; len(houses):             sel_check = True</pre>		
<pre>if sub_total &gt; 100000:     total_fees += 3000+(sub_total-100000) * 0.2 else:     total_fees += sub_total *0.3</pre>	<pre>if sub_total &gt; 100000:     total_fees += 3000 + (sub_total - 100000) * 0.02 else:     total_fees += sub_total * 0.03</pre>	
<pre>print('Customer Receipt\n\n FORENAME:{} SURNAME: {} PROPERTY COST: {} WITH STAMP DUTY: {}'.format(forename, surname, property_cost, stamp_duty)) print('\nTRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES DATABASE\n') print (houses[select]) del houses[select]</pre>	<pre>print(houses[select - 1])</pre>	
<pre>if menuselection == 1:     return_stock if menuselection == 2:     region_search if menuselection ==3:     house_sale if menuselection==4:     show_sales</pre>	<pre>if menuselection == 1:     return_stock() if menuselection == 2:     region_search() if menuselection == 3:     house_sale() if menuselection == 4:     add_house() if menuselection == 5:     show_sales() if menuselection == 6:</pre>	

The test plan/log should also contain inclusion of tests that show that areas of the program that appear to be coded correctly have been tested to ensure outputs are correct and the program is robust. These may include (but not limited to):

- Positive and negative values for number of products in stock
- Non numerical values for number each item
- Incorrect inputs e.g. products not in the list, numerical values entered instead of name of product
- Manual calculations of totals to check program is outputting correct values for totals
- Adding new stock to the product array
- Testing the different options available to the user through the menu

A limited understanding of program requirements would be typified by only identifying and fixing the functional errors that would be highlighted by the IDE, but would not identify and fix logical errors. The number of errors identified is not a hurdle between Band 2 and 3, the discriminator is the quality and appropriateness of the tests selected, and the level of understanding shown of the process

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1-2	3-5	6-8
Use of testing to identify defects	No rewardable material	Tests selected show a basic understanding of the identified program requirements.  Test log includes some appropriate test data.  Testing has identified some error in the code provided.	Tests selected show a good understanding of the identified program requirements.  Test log includes a good a range of normal, erroneous and extreme data.  Testing has identified most errors in the code provided.	Tests selected show a thorough and detailed understanding of the identified program requirements.  Test log includes a comprehensive range of normal, erroneous and extreme data.  Testing has comprehensively identified the errors in the code provided.
		<b>1</b>	<b>2-3</b>	<b>4</b>
Understanding of the testing process		Test log shows a basic understanding of how errors/problems were identified and how they were rectified.	Test log shows a good understanding of how errors/problems were identified and how they were rectified.	Test log shows a thorough and detailed understanding of how errors/problems were identified and how they were rectified.
		<b>1-3</b>	<b>4-6</b>	<b>7-9</b>
The solution		Code has some functionality, but significant errors still persist.  Changes made apply some precise logic and programming structures which would result in some correct outcomes.	Code is mostly functional, but some minor errors still persist.  Changes made apply mostly precise logic and programming structures which would result in mostly correct outcomes.	Code is fully functional.  Changes to code apply fully precise logic and programming structures throughout which would result in consistently correct outcomes

### Task 3- Designing a solution

#### Indicative content and marker guidance

##### General guidance:

- Algorithm designs should demonstrate decomposition of the problem into simpler and more understood primitives. A decomposition diagram is not required, but if provided mark for decomposition, otherwise for consideration of decomposition in design documents e.g. flowchart and pseudocode
- The design should provide high level coverage of the process as well as identify reusable components
- Detailed algorithms (pseudocode) do not need to be provided for ALL repeated processes. For example if the process for calculating sales for regions is very similar, the learner would not need to provide algorithms for each region, rather they should show how reusable code and may provide some additional annotation to explain the process as necessary.
- Good decomposition will show all the necessary processes and sub-processes that make up the main problem these might include:
  - Importing data from the CSV file
  - Taking input to select region
  - Iterating through data for specified data
  - Taking a date range specified for data
  - A calculation related to increase in value
  - Generation of data frames and graphs
  - Aspects related to validating user input

##### Some general characteristics of a good algorithm that may be demonstrated are:

- the steps are clearly defined
- each step is uniquely defined and should depend on the input and the result of the preceding steps
- receives input, selection of regions and dates, as well as regions – inputs for user interface should also be considered
- produces appropriate type of output e.g. screen display, return value or return list, which results are required, what happens if no results can be computed maybe error
- Sensible naming conventions for variables and processes
- Use of key words, symbols, hierarchies, and structures as appropriate to the chosen method to represent the algorithm (i.e., pseudocode or flowchart)

##### Scenario specific characteristics may include:

- Suitable logic for selection of regions and dates to show value increases
- Suitable calculations to show increase in values over time
- Links to CSV to get data.
- Sensible use of CSV or run-time data structure (e.g data frame) to hold different parts of the data for processing e.g.
  - List to hold menu options, or predefined date ranges
  - New data frame to hold data for the selected date range to aid calculations
- Understanding of given data such as:
  - Use of header row in CSV to required data (e.g. region, house type)
  - Need to convert dates to a usable date format
- Simplification for user e.g. choose a number from menu rather than type the region in full

##### Example detailed algorithms for repeated processes:

Note – these are intended to be indicative of the types of algorithms that may be presented. These **do not** show all processes. Accept any responses that provide logically correct outcomes/solutions

**Get and validate Region choice**

```
WHILE not_valid_input_flag = TRUE
    SEND "Please enter the number of the region you wish to view TO DISPLAY"

    SET region_list TO [<list of regions>]

    RECEIVE region_choice (integer) FROM KEYBOARD

    IF region_choice NOT integer THEN:
        SEND "That is not a valid choice" TO DISPLAY
        SET not_valid_input_flag TO TRUE
    ELSE IF region_choice <1 OR >6 THEN:
        SEND "That is not a valid choice" TO DISPLAY
        SET not_valid_input_flag TO TRUE
    ELSE:
        SET region_choice TO region_list [int(region_choice)-1]
        RETURN
region_name
```

**Get and validate date range**

```
SET not_valid_flag TO TRUE

while flag == TRUE
    SEND "Please enter a start date for your time range (Jan-20) TO DISPLAY"

    RECEIVE input (STRING) FROM KEYBOARD

    try:
        SET input (STRING) TO input (DATE1)
    except:
        SEND "Sorry, you did not enter a valid date" TO DISPLAY
        flag = True
    else:
        SET start_date TO input (DATE1)
        RETURN start_date
```

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1-2	3-5	6-8
<b>Decomposition of the problem</b>	No rewardable material	Basic or superficial decomposition of the identified problems that superficially cover the required: <ul style="list-style-type: none"> <li>• inputs</li> <li>• processes</li> <li>• outputs.</li> </ul>	Mostly detailed decomposition of the identified problems that sufficiently cover the required: <ul style="list-style-type: none"> <li>• inputs</li> <li>• processes</li> <li>• outputs.</li> </ul>	Thorough and detailed decomposition of the identified problems that comprehensively cover the required: <ul style="list-style-type: none"> <li>• inputs</li> <li>• processes</li> <li>• outputs.</li> </ul>
		1-2	3-4	5-6
<b>Application of logical thinking and conventions</b>		Algorithms would produce some correct outcomes as a result of: <ul style="list-style-type: none"> <li>• some precise logic</li> <li>• some appropriate structure and sequence which is likely to be inefficient.</li> </ul>	Algorithms would produce mostly correct outcomes as a result of: <ul style="list-style-type: none"> <li>• mostly precise logic</li> <li>• appropriate structure and sequence but which may lack efficiency.</li> </ul>	Algorithms would produce correct outcomes as a result of: <ul style="list-style-type: none"> <li>• precise logic</li> <li>• Appropriate and efficient structure and sequence.</li> </ul>
		Some use of accepted conventions.	Mostly appropriate use of accepted conventions though some minor inconsistencies may still exist.	Appropriate and consistent use of accepted conventions.
		1	2	3
<b>Communication of the design</b>		Superficial communication of the design uses technical language which is only sometimes appropriate for the audience.	Adequate communication of the design uses technical language which is mostly appropriate for the audience.	Effective communication of the design uses technical language which is appropriate for the audience.



**Task 4a- Developing the solution****Indicative content and marker guidance**

## The solution

- Provides a developed coded solution that utilises the given code and adds the additional functionality as stated in the requirements.
- Integration of existing code may include:
  - 'import' function to pull code as a whole in when needed (note given code and learner code will need to be in the same folder)
  - Integration into learners own code base as a function
- The solution will be well structured and modular in nature – clearly see subsections this may be separated modules or the use of Procedures, Functions or classes
- Code will be annotated to aid future maintenance of the system
- Data/information should be output in a meaningful way to the user. This may include use of a data-frame, graph and/or text based summary
- Output data should show consideration of different sale people over time but this may be interpreted in slightly different ways. Such as:
  - Higher level responses will make better use of patterns and trends over time and allow the user to select regions and house types, as well as dates in different combinations to give more meaningful data analysis.
  - lower levels responses may only extract based on a single criterion e.g. a single region, making use of a hard coded date range instead of user input
  - Learners make use of tabulated formats or line graphs to display this information in a clearer way.
  - Lower level responses may extract some meaningful information but may only output in an unsorted fashion or may not give the user sufficient choice
- There are different ways that this task can be interpreted so the characteristics of higher level responses will show a greater discrimination of the data to be extracted.
- Code should be robust, typical errors that may be accounted for in this scenario include negative values, non-numerical characters, entering a choice not provided in the menu, misspelling of region names/wrong capitalization

## Possible areas included that contribute to 'user experience':

- Outputs are meaningful and make sense to the user e.g. outputs are accompanied by meaningful text to contextualise them
- Simplification of input processes e.g. use of numbers in a menu rather than writing the full region names.
- Creation of new columns in a dataframe to calculate values
- Numerical values are rounded to a sensible number of decimal places and currency symbol added as appropriate
- Use of visualization e.g. bar graphs to show values over time
- Helpful messages and robust input handling

## Security consideration relating to the solutions could include:

- Avoiding global variables, passing data back from functions
- Avoiding the use of a single generated dataframe to ensure security of the data – good practice to generate a new data frame within a function
- error handling: If the system crashes, is any data returned in the error message

**Example code snippets for parts of the solution:**

Note – these are intended to be indicative of the types of re that may be presented. These **do not** show all processes. Accept any responses that provide logically correct outcomes/solutions

**Developing user menus to incorporate functionality**

```
print("\t\t****Welcome to the Dashboard****")

print('1 Return all current data')
print('2 Return data for a specific region')
print('3 Return data based on property types in a region')
print('4 find region with greatest increase')
```

**Showing regional totals**

```
def region_validation():
    region = input("Enter region name:")
    return region.capitalize()

def start_date():
    startdate = input("PLEASE ENTER A START DATE AS MONTH-YEAR e.g. JAN-20")
    return startdate.capitalize()

def end_date():
    enddate = input("PLEASE ENTER AN END DATE AS MONTH-YEAR e.g. JAN-20")
    return enddate.capitalize()
```

**Generating data based on user inputs**

```
for x, y in enumerate(df["Region Code"]):
    if y == region_codes[selected_region]:
        property_indexes.append(x + 2)

for x in property_indexes:
    if df.iloc[x - 2]["Property Type"] == selected_property:
        print(df.iloc[x - 2])
```



Assessment focus	Band 0	Band 1	Band 2	Band 3	Band 4
	0	1-2	3-4	5-6	
<b>Functionality</b>	No rewardable material	The solution implements code with some functionality but some major errors still persist.	The solution implements mostly functional code but code may lack efficiency and some minor errors still persist.	The solution implements functional and efficient code throughout.	
		<b>1</b>	<b>2</b>	<b>3</b>	
<b>Logic and programming structures</b>		The code uses some precise logic and programming structures which would result in some correct outcomes.	The code uses mostly precise logic and programming structures which would result in sufficiently correct outcomes.	The code uses precise logic and programming structures throughout which would result in consistently correct outcomes.	
		<b>1</b>	<b>2</b>	<b>3</b>	
<b>Robustness</b>		The code handles some common user errors	The code handles most common user errors	The code thoroughly handles common, and most unexpected, user errors	
		<b>1-2</b>	<b>3-4</b>	<b>5-6</b>	
<b>Security</b>		The code mitigates against some common vulnerabilities as a result of some effective application of secure coding practices.	The code mitigates against most relevant vulnerabilities through mostly effective application of secure coding practices	The code thoroughly mitigates against relevant vulnerabilities through effective application of secure coding practices	
		<b>1-2</b>	<b>3-4</b>	<b>5-6</b>	<b>7-8</b>
<b>Code organisation</b>	The code is partially maintainable by a third party but would present significant difficulties through the use of: <ul style="list-style-type: none"> <li>inconsistent naming conventions</li> <li>limited logical organisation</li> <li>limited informative commenting</li> </ul>	The code is partially maintainable by a third party but would present some minor difficulties through the use of: <ul style="list-style-type: none"> <li>some consistent naming conventions</li> <li>some logical organisation</li> <li>some informative commenting</li> </ul>	The code is maintainable by a third party and would present only a few minor difficulties through the use of: <ul style="list-style-type: none"> <li>mostly consistent naming conventions</li> <li>mostly logical organisation</li> <li>mostly informative commenting</li> </ul>	The code is easily maintainable by a third party through the use of: <ul style="list-style-type: none"> <li>consistent and appropriate naming conventions</li> <li>fully logical organisation</li> <li>highly informative commenting</li> </ul>	



	0	1-2	3-4	5-6	7-8
<b>User experience</b>	No rewardable material	Basic user experience is provided through limited effective use of: <ul style="list-style-type: none"> <li>input handling</li> <li>user guidance and error messages</li> <li>outputs</li> </ul>	Adequate user experience is provided through somewhat effective use of: <ul style="list-style-type: none"> <li>input handling</li> <li>user guidance and error messages</li> <li>outputs</li> </ul>	Good user experience is provided through mostly effective use of: <ul style="list-style-type: none"> <li>input handling</li> <li>user guidance and error messages</li> <li>outputs</li> </ul>	Excellent user experience is provided through consistently effective use of: <ul style="list-style-type: none"> <li>input handling</li> <li>user guidance and error messages</li> <li>outputs</li> </ul>

**Task 4b- Reflective evaluation****Indicative content and marker guidance**

Indicative content will vary according to the approach learners have taken in task 4a and the effectiveness of the solution they created.

Generic features of effective evaluations are likely to include:

- the extent to which the solution meets the:
  - requirements of the set task brief
  - needs of the users.
- a justification of how the solution could be further developed/enhanced.
- specific examples from the solution to support points made
- contextualisation of any points made and explaining what they did and justifying why.

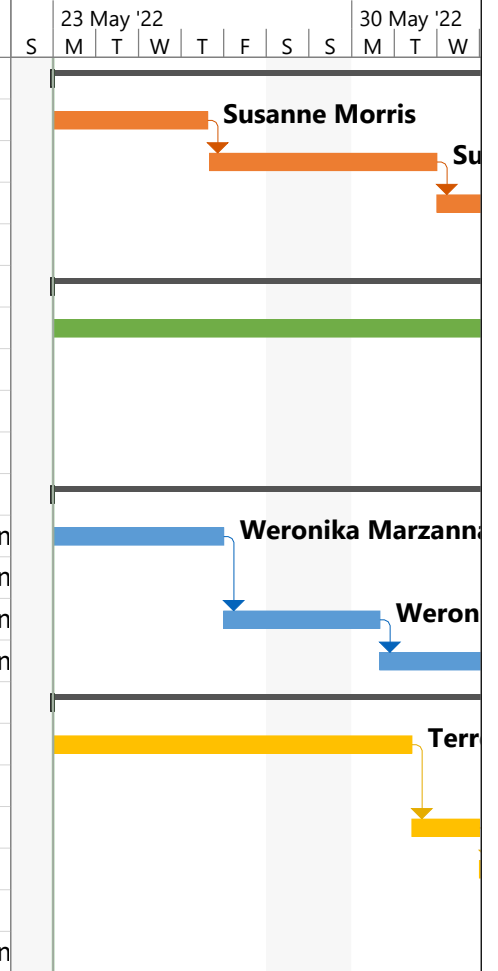
Contextualisation for this scenario may include:

- Choice of data output (e.g. text, table, graph type) – which is most suitable for showing value of properties over time/region with the highest increase in property value
- Rounding vs truncation for calculations and consideration of data types to specify number of decimal places for the calculation output
- How the brief was interpreted – such as how regions were selected by the user, were specific date ranges entered
- How existing code was integrated
- Choice of libraries/functions to get required data
- How most region with the highest increase in property value has been determined e.g. have user inputs for dates been considered, how has the information been grouped/sorted
- How data was extracted (e.g. use directly from the csv file, use of subsets of data, how the choice of data frame or datafile impacted on search/extraction)
- Use of variables (global vs local, passing data between functions)
- Input error handling – e.g. why they might have excluded text or negative values, menu options etc

Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1-2	3-4	5-6
Programming outcomes	No rewardable material	Judgements reached are somewhat supported showing a superficial or basic understanding of how well the solution met the: <ul style="list-style-type: none"> <li>requirements of the set task brief</li> <li>needs of the users.</li> </ul>	Judgements reached are mostly well supported showing a good understanding of how well the solution met the: <ul style="list-style-type: none"> <li>requirements of the set task brief</li> <li>needs of the users.</li> </ul>	Judgements reached are comprehensively well supported showing a thorough and detailed understanding of how well the solution met the: <ul style="list-style-type: none"> <li>requirements of the set task brief</li> <li>needs of the users.</li> </ul>
		<b>1</b>	<b>2</b>	<b>3</b>
Future Developments		A superficial or simplistic rationale is provided for what future developments should be implemented.	A good rationale which is reasonably well supported is provided for what future developments should be implemented.	A convincing and well-supported rationale is provided for what future developments should be implemented.

'T-LEVELS' is a registered trade mark of the Department for Education.  
 'T Level' is a registered trade mark of the Institute for Apprenticeships and Technical Education.  
 The T Level Technical Qualification is a qualification approved and managed by the Institute for Apprenticeships and Technical Education.  
 'Institute for Apprenticeships & Technical Education' and logo are registered trade marks of the Institute for Apprenticeships and Technical Education.  
 Pearson Education Limited is authorised by the Institute for Apprenticeships and Technical Education to develop and deliver this Technical Qualification.  
 All the material in this publication is copyright © Pearson Education Limited 2022

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		<b>Hardware and Networking</b>	<b>15.14 days</b>	<b>Mon 23/05/22</b>	<b>Thu 09/06/22</b>		
2		Upgrade Infrastructure	30 hrs	Mon 23/05/22	Thu 26/05/22		Susanne Morris
3		Install and configure server	25 hrs	Thu 26/05/22	Tue 31/05/22	2	Susanne Morris
4		Installation of hardware	28 hrs	Wed 01/06/22	Mon 06/06/22	3	Susanne Morris
5		Testing of hardware	22 hrs	Mon 06/06/22	Thu 09/06/22	4	Susanne Morris
6		<b>Module 1: Backend Database</b>	<b>22.14 days</b>	<b>Mon 23/05/22</b>	<b>Fri 17/06/22</b>		
7		Develop	110 hrs	Mon 23/05/22	Thu 09/06/22		Ayaru Thapa
8		Deploy	6 hrs	Thu 16/06/22	Fri 17/06/22	10	Ayaru Thapa
9		Unit Tests	14 hrs	Thu 09/06/22	Mon 13/06/22	7	Ayaru Thapa
10		Fixing and regression for minor fault	25 hrs	Mon 13/06/22	Thu 16/06/22	9	Ayaru Thapa
11		<b>Module 2: Sales records</b>	<b>11.14 days</b>	<b>Mon 23/05/22</b>	<b>Fri 03/06/22</b>		
12		Develop	33 hrs	Mon 23/05/22	Fri 27/05/22		Weronika Marzan
13		Deploy	6 hrs	Fri 03/06/22	Fri 03/06/22	15	Weronika Marzan
14		Unit Tests	14 hrs	Fri 27/05/22	Mon 30/05/22	12	Weronika Marzan
15		Fixing and regression for minor fault	25 hrs	Mon 30/05/22	Thu 02/06/22	14	Weronika Marzan
16		<b>Module 3: User interface</b>	<b>14.29 days</b>	<b>Mon 23/05/22</b>	<b>Wed 08/06/22</b>		
17		Develop	50 hrs	Mon 23/05/22	Tue 31/05/22		Terrence Bull
18		Deploy	6 hrs	Tue 07/06/22	Wed 08/06/22	29	Nasira Amina
19		Unit Tests	14 hrs	Tue 31/05/22	Wed 01/06/22	17	Terrence Bull
20		Fixing and regression for minor fault	25 hrs	Thu 02/06/22	Tue 07/06/22	19	Terrence Bull
21		<b>Module 4: Data analysis</b>	<b>20.14 days</b>	<b>Fri 03/06/22</b>	<b>Wed 29/06/22</b>	<b>11</b>	
22		Develop	110 hrs	Fri 03/06/22	Thu 23/06/22		Weronika Marzan



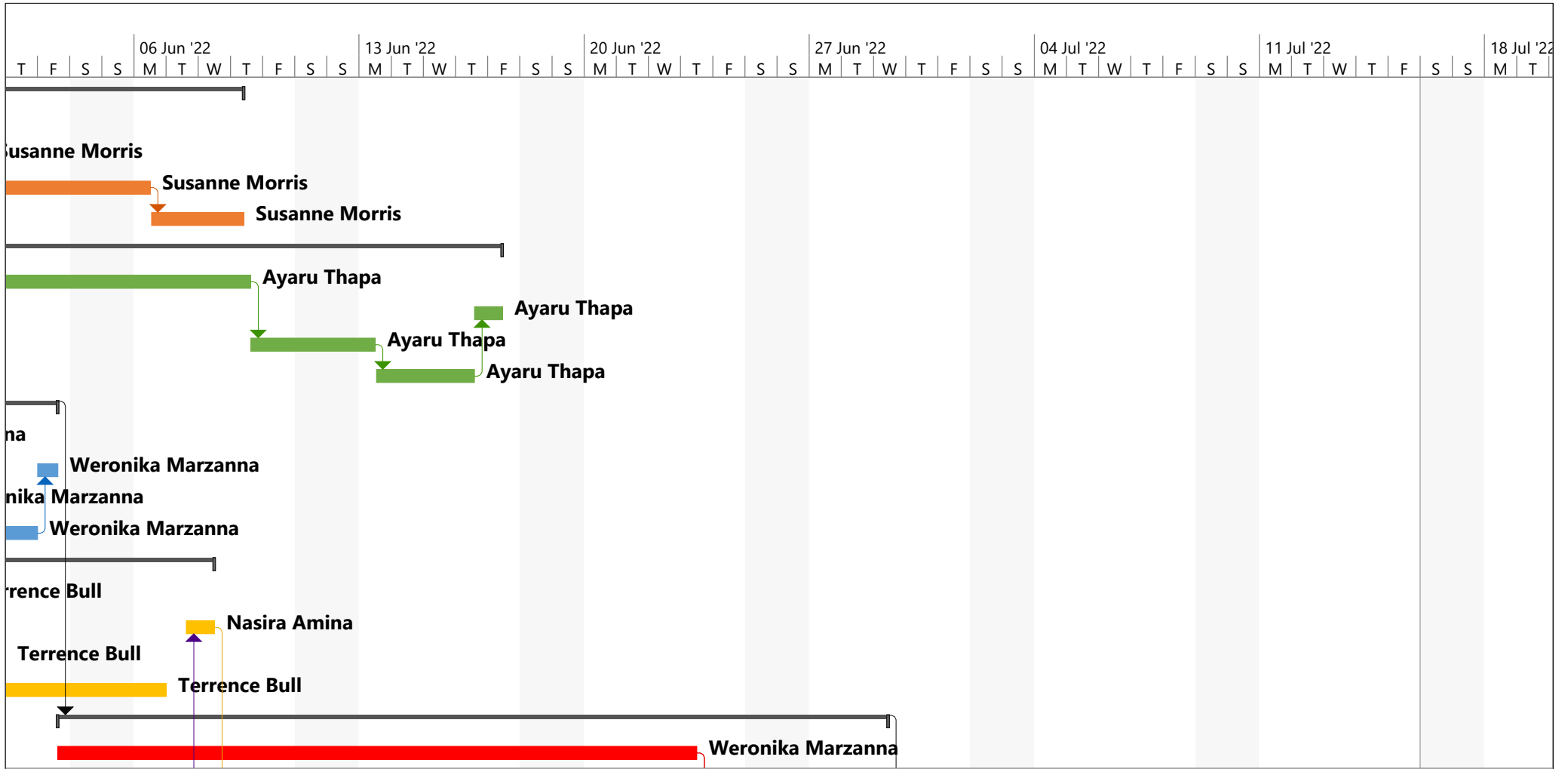
Project: Task1GanttChart\_LL-00  
Date: Mon 23/05/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names	23 May '22							30 May '22				
								S	M	T	W	T	F	S	S	M	T	W	
23		Deploy	6 hrs	Tue 28/06/22	Wed 29/06/22	25	Weronika Marzan												
24		Unit Tests	14 hrs	Thu 23/06/22	Mon 27/06/22	22	Nasira Amina												
25		Fixing and regression for minor fault	25 hrs	Thu 23/06/22	Tue 28/06/22	22	Weronika Marzan												
26		<b>Module 5: Customer Support</b>	<b>18.71 days</b>	<b>Mon 23/05/22</b>	<b>Tue 14/06/22</b>														
27		Develop	80 hrs	Mon 23/05/22	Fri 03/06/22		Nasira Amina												
28		Deploy	6 hrs	Mon 13/06/22	Tue 14/06/22	30	Nasira Amina												
29		Unit Tests	14 hrs	Mon 06/06/22	Tue 07/06/22	27	Nasira Amina												
30		Fixing and regression for minor fault	25 hrs	Wed 08/06/22	Mon 13/06/22	18	Nasira Amina												
31		<b>Testing</b>	<b>9.43 days</b>	<b>Wed 29/06/22</b>	<b>Mon 11/07/22</b>	<b>21</b>													
32		Create a test plan	7 hrs	Wed 29/06/22	Thu 30/06/22		Ayaru Thapa												
33		Integration Testing (modules 2 and 4)	30 hrs	Wed 29/06/22	Tue 05/07/22		Weronika Marzan												
34		Integration Testing (modules 1 and 3)	30 hrs	Wed 29/06/22	Tue 05/07/22		Nasira Amina												
35		Integration Testing (module 5)	15 hrs	Fri 01/07/22	Mon 04/07/22	37	Terrence Bull												
36		Fixing and regression for major fault	36 hrs	Tue 05/07/22	Mon 11/07/22	34	Nasira Amina												
37		User/acceptance testing	13 hrs	Wed 29/06/22	Thu 30/06/22		Terrence Bull												
38		<b>Training</b>	<b>4.29 days</b>	<b>Mon 11/07/22</b>	<b>Fri 15/07/22</b>	<b>31</b>													
39		Staff training	30 hrs	Mon 11/07/22	Fri 15/07/22		Terrence Bull												
40		User training	15 hrs	Mon 11/07/22	Wed 13/07/22		Ayaru Thapa												

Project: Task1GanttChart\_LL-00  
Date: Mon 23/05/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			



Project: Task1GanttChart\_LL-00  
Date: Mon 23/05/22

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			





**Weronika Marzanna****£100 per hour**

Task	Cost
Develop module 2	£3,300
Deploy module 2	£600
Unit testing module 2	£1,400
Fixing and regression for minor faults (module 2)	£2,500
Develop module 4	£11,000
Deploy module 4	£600
Fixing and regression for minor faults (module 4)	£2,500
Integration Testing (modules 2 and 4)	£3,000

**Total Cost****£24,900****Terrence Bull****£45 per hour**

Task	Cost
Develop module 3	£2,250
Unit testing module 3	£630
Fixing and regression for minor faults (module 3)	£1,125
Integration Testing (module 5)	£675
User/acceptance testing	£585
Staff training	£1,350

**Total Cost****£6,615****Nasira Amina****£35 per hour**

Task	Cost
Deploy module 3	£210
Unit testing module 4	£490
Develop module 5	£2,800
Deploy module 5	£210
Unit testing module 5	£490
Fixing and regression for minor faults (module 5)	£875
Integration testing (modules 1 and 3)	£1,050
Fixing and regression for major faults	1260

**Total Cost****£7,385****Susanne Morris****£70 per hour**

Task	Cost
Upgrade Infrastructure	£2,100
Install and configure server	£1,750

Installation of hardware	£1,960
Testing of hardware	£1,540

**Total Cost**

£7,350

**Ayaru Thapa**

£75 per hour

Task	Cost
Develop module 1	£8,250
Deploy module 1	£450
Unit testing module 1	£1,050
Fixing and regression for minor faults (module 1)	£1,875
Create a test plan	£525
User training	£1,125

**Total Cost**

£13,275

**Total Cost**

Task	Cost
Purchasing of hardware	£3,300
Training for staff	£600
Software development company base fee	£1,400
Software liscensing (Option 1: Single fee)	£2,500
Upgrades to company infrastructure	£11,000
Server Upgrade	£600
Project Lead's pay (£140 per day x 33 working days)	£4,620
Integration Testing (modules 2 and 4)	£3,000
Weronika Marzanna's pay	£24,900
Terrence Bull's pay	£6,615
Nasira Amina's pay	£7,385
Susanne Morris's pay	£7,350
Ayaru Thapa's pay	£13,275

6 weekends

**Total Cost**

£86,545

**Running Annual Costs**

Task	Cost
Maintaining servers and equipment	£2,100

**Total Cost**

£2,100

## Revenue

Task	Cost
Current	£912,000
First Year	£939,360
Second Year	£966,720
Third Year	£994,080
Total With Solution	£2,900,160
Total Without Solution	£2,736,000
<b>Total additional revenue</b>	<b>£164,160</b>

## Total Profit After 3 Years

Task	Cost
Total additional revenue	£164,160
Running Annual Costs (after 3 years)	-£6,300
Cost of Solution	-£86,545
<b>Total Profit</b>	<b>£71,315</b>

# Rationale

In this document I will explain my planning approach and justify the decisions that I have made.

## Weronika Marzanna

I have allocated the resource of Senior Software Engineer Weronika Marzanna by assigning her to develop module 2 (sales records) and module 4 (data analysis). Weronika is the perfect candidate for these modules due to her ability to “develop modules around data analysis and sales to a consistently high standard”.

## Terrence Bull

I have allocated the resource of Junior Software Engineer Terrence Bull to developing module 3 (user interface). Terrence is the perfect candidate for developing this module due to his experience working as a user interface designer. Terrence is also the appropriate choice for staff training because of his experience working “in a staff development training role”. I have also assigned Terrence to User/acceptance testing because of his “certification in multiple methods of software testing”.

## Nasira Amina

I have allocated the resource of Junior Software Engineer Nasira Amina by assigning her to develop module 5 (customer support). Nasira is an appropriate choice because whilst multiple members of the team would be capable of completing this task, Nasira is the most inexpensive. I also assigned Nasira to just the deployment of module 3 (user interface), I did this because Terrence Bull (the resource assigned to the other aspects of module 3) has “limited experience in deployment of software”. Because of this I figured out the schedule so that Nasira could deploy it instead of Terrence in order to make sure it is done by someone who doesn't have deployment as a known weakness.

## Susanne Morris

I have allocated the resource of Hardware and Networking Technician Susanne Morris by using her “experience of updating and upgrading a range of infrastructure” when assigning her to upgrading the infrastructure. Her experience as a Hardware and Networking Technician will also make her the appropriate choice for installing and configuring the server as well as installing the hardware. Additionally, Susanne is the ideal choice for the testing of hardware due to her “experience of developing testing plans and hardware testing”.

## Ayaru Thapa

I have allocated the resource of Database Engineer Ayaru Thapa by assigning her to develop module 1 (backend database). Ayaru is the perfect fit for this task due to her 14 years of experience working as a database engineer. I have also assigned Ayaru to creating a test plan because of her “certification in multiple methods of software testing”. Finally, Ayaru is the perfect fit for User training because of her “ability to explain complicated processes in non-technical language”.

## Choice of Software License

I was given the choice between paying a single fee of £1,500 for software licensing (with included technical support for three years) or £250 annually (without technical support). I decided to choose option 1 (the single fee) because even though it is initially more expensive than the annual fee, after 6 years, the cost of the annual fee will have become greater and will continue to grow over time. Additionally, the three years of technical support could potentially be worth more than the fee itself if a problem were to arise and the company would otherwise have to spend time and resources in order to identify and solve the problem.

# Rationale

## Timeframe

The company requested that this project be completed in 12 weeks, my plan for this project fits within that timeframe by completing it in under 7 weeks or 33 working days.

## Is The Project Profitable?

As shown in my resource cost plan, the development of the solution will gain the company an extra £164,160 in revenue after the first three years of its deployment which after subtracting the annual cost of server/equipment maintenance as well as the cost of the solution itself equals £71,315. This proves that developing the solution would be profitable for the business and therefore worthwhile developing.

```

houses = [['LONDON','Terraced', 3, 735000], ['CARDIFF', 'Semi-Detached',
2, 100000], ['LEEDS', 'Terraced', 3,245000],['LONDON','Semi-Detached', 1,
240000]]
sales = []
ourregions = ['LONDON', 'LEEDS', 'CARDIFF', 'BRISTOL']
property_types = ['TERRACED', 'SEMI-DETACHED','DETACHED']

def return_stock():
    print("CURRENT HOUSES FOR SALE \n\n REGION - HOUSE TYPE - BEDROOMS -
COST")
    for i in houses:
        print (i)

def unique_regions():
    unique_list = []
    existing_regions = [item[0] for item in houses]
    for x in existing_regions:
        if x not in unique_list:
            unique_list.append(x)
    print(unique_list)

def region_search():
    print("Available Regions")
    unique_regions()
    r_check = False

    while not r_check:
        region_select= input("Please enter region: ").capitalize()

        for x in houses:
            if region_select.upper() == x[0].upper():
                r_check = True
                if x[0] == region_select.upper():
                    print(x)

        if r_check == False:
            print("Entered region is not valid")

def show_sales():
    if len(sales) > 0:
        print("Forename Surname Property cost Total")
        for i in sales:
            print(i)
    else:
        print('no sales')

def house_sale():
    sale = []
    customer_forename = ""
    while not customer_forename:
        customer_forename = input('Please enter customer forename: ')
    customer_surname = ""
    while not customer_surname:

```

```

        customer_surname = input('Please enter customer surname: ')
for i, item in enumerate(houses, 1):
    print(i, item)

sel_check = False
while not sel_check:
    try:
        select = int(input('Please select a purchase: '))
        if select > 0 and select < len(houses):
            sel_check = True
        else:
            print("Please enter a number between 1 and " +
str(len(houses)))

    except:
        print('ERROR PLEASE ENTER A VALID PROPERTY')

sub_total = houses[select-1][3]
print(sub_total)
total_fees = 0

if sub_total > 100000:
    total_fees += 3000 + ((sub_total-100000) * 0.02)
else:
    total_fees += sub_total *0.03

final_total = sub_total+total_fees
sale.append(customer_forename)
sale.append(customer_surname)
sale.append(round(sub_total,2))
sale.append(round(final_total,2))
sales.append(sale)

print('Customer Receipt\n\nFORENAME:{} SURNAME: {} PROPERTY COST:
£{} WITH STAMP DUTY: £{}'.format(*sales[-1]))
print('\nTRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES
DATABASE\n')
print (houses[select-1])
del houses[select-1]

def add_property():
    house = []
    print(ourregions)
    while True:
        region = input("Please enter the region of this property:
").upper()
        if region in ourregions:
            house.append(region)
            break
        else:
            print("You can only enter one of the regions shown above")

print("House types: " + ", ".join(property_types))
while True:
    house_type = input("Please enter the house type: ").upper()

```

```

        if house_type in property_types:
            house.append(house_type)
            break
        else:
            print("You can only enter one of the house types shown
above")

while True:
    bedrooms = input("Please enter the number of bedrooms: ")
    if bedrooms.isnumeric() and 0 < int(bedrooms):
        house.append(int(bedrooms))
        break
    else:
        print("You can only enter a number (and it must be above 0)")

while True:
    cost = input("Please enter the house's cost: ")
    cost = cost.replace("£", "")
    if cost.isnumeric() and 0 < int(cost) < 10000000000000:
        house.append(int(cost))
        break
    else:
        print("You must enter the house's cost as a positive
integer")
print(house)
while True:
    choice = input("Does this look good? (Y/N)\n").upper()
    if choice in ["Y", "N", "YES", "NO"]:
        break
if choice in ["Y", "YES"]:
    houses.append(house)
    print("Property added!")
else:
    add_property()

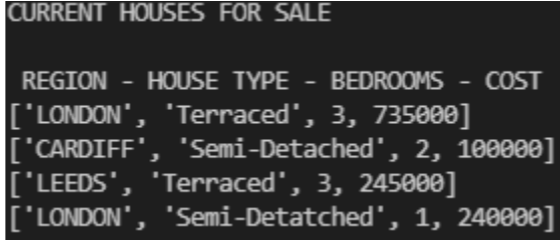
while True:
    menu_selection = input(" WELCOME TO THE NEWHAVEN DASHBOARD \n\n
Please select from the following menu options \n\n"
        " 1: View current houses on market \n 2:
Search for available houses in a region \n 3: Record"
        " a sale \n 4: Add a new property for sale
\n 5: Show Sales \n 6: Exit\n")

    if menu_selection == "1":
        return_stock()
    elif menu_selection == "2":
        region_search()
    elif menu_selection == "3":
        house_sale()
    elif menu_selection == "4":
        add_property()
    elif menu_selection == "5":
        show_sales()
    elif menu_selection == "6":
        break
    else:
        print("Input must be between 1 and 6")

```



## Test 2 - Test Log

Description of test	Test data to be used (if required)	Expected outcome	Actual outcome	Comments and intended actions
Attempt to run the program		Program will start without error	<p>Test 1: Invalid syntax on line 2</p> <p>Test 2: Invalid syntax on line 59</p> <p>Test 3: Program ran without causing an error</p>	<p>Test 1: The declaration of houses array is missing some commas between values</p> <p>Test 2: Input statement has an extra closing bracket</p> <p>Test 3: Program worked as expected</p>
Testing the menu: Option 1 (View current houses on the market)	menuselection = 1	The program will accept the value and take the user to viewing current houses on the market	<p>Test 1: The program stayed on the main menu</p> <p>Test 2: The program displayed the current available houses before returning to the main menu</p>  <pre> CURRENT HOUSES FOR SALE  REGION - HOUSE TYPE - BEDROOMS - COST ['LONDON', 'Terraced', 3, 735000] ['CARDIFF', 'Semi-Detached', 2, 100000] ['LEEDS', 'Terraced', 3, 245000] ['LONDON', 'Semi-Detached', 1, 240000]                     </pre>	<p>Test 1: The program attempts to call the functions from the menu without using brackets</p> <p>Test 2: Program worked as expected</p>
Testing the menu: Option 2 (Search for available houses in a region)	menuselection = 2	The program will accept the value and take the user to searching for available houses in a region	The program accepted the value and navigated the user to the correct section of code	Program worked as expected
Testing the menu: Option 3 (Record a sale)	menuselection = 3	The program will accept the value and take the user to recording a sale	The program accepted the value and navigated the user to the correct section of code	Program worked as expected
Testing the menu: Option 4 (Add a new property for sale)	menuselection = 4	The program will accept the value and take the user to adding a new property for sale	<p>Test 1: Error on line 41, 'sold' is not defined</p> <p>Test 2: The program took the user to the show_sales function (meant for option 5) and the option for adding a new property for sale doesn't exist in the code</p> <p>Test 3: The program accepted the value and navigated the user to the correct section of code</p>	<p>Test 1: 'sold' needed to be changed to 'sales'</p> <p>Test 2: A new function for adding a property needs to be developed and this option should call that function instead of the show_sales function.</p> <p>The function I added will ask the user to enter the region, house-type, bedroom number and cost of the house and will add it to the houses array. My code displays the information to the user in a meaningful way, handles user errors appropriately and gives them the option to start again and re-enter the details if they aren't happy with them.</p> <p>Test 3: Program worked as expected</p>

Testing the menu: Option 5 (Show Sales)	menuselection = 5	The program will accept the value and take the user to showing the sales	Test 1: The program stayed on the main menu  Test 2: The program accepted the value and navigated the user to the correct section of code	Test 1: The program doesn't include the if statements in order to accommodate options 5 and 6  I added these options and also changed most of the existing if statements into elifs in order to fit better with general coding practices.  Test 2: Program worked as expected
Testing the menu: Option 6 (Exit)	menuselection = 6	The program will accept the value and exit the program	The program accepted the value and quit the program	Program worked as expected
Testing the menu: Putting in a number which is out of range of the options	menuselection = 100	The program will give the user an error message before asking them to enter the input again	Test 1: The program handled the user error but did not give any error message  Test 2: The program gave the user an error message before asking them to enter the input again <code>Input must be between 1 and 6</code>	Test 1: I added an else statement in order to give the user an error message if the value they entered is out of range <code>else: print("Input must be between 1 and 6")</code>  Test 2: Program worked as expected
Testing the menu: Entering a non-numeric input	menuselection = three	The program will give the user an error message before asking them to enter the input again	Test 1: Program crashed trying to convert a non-numeric string into an integer  Test 2: The program gave the user an error message before asking them to enter the input again <code>Input must be between 1 and 6</code>	Test 1: The program needs to be able to handle user value errors by not converting any input to an integer, the easiest way to do it in this case is to change the if statements to check for strings instead of integers. I also changed the variable name to menu_selection to better fit with general coding standards. <code>menu_selection = input( if menu_selection == "1": return_stock() elif menu_selection == "2": region_search()</code>  Test 2: Program worked as expected
Testing Option 2: Are the Available regions correct?		The available regions should be: LONDON, CARDIFF and LEEDS	Test 1: Returned an empty list  Test 2: <code>Available Regions ['LONDON', 'CARDIFF', 'LEEDS']</code>	Test 1: The code for creating a unique list of regions doesn't work as intended Fix: <code>if x not in unique_list:</code>  Test 2: Program worked as expected
Testing Option 2: Entering a valid region	region_select = LONDON	The program should accept the value and show the relevant properties (2 in this case)	<code>['LONDON', 'Terraced', 3, 735000] ['LONDON', 'Semi-Detached', 1, 240000]</code>	Program worked as expected
Testing Option 2: Entering a valid region in lowercase	leeds	The program should accept the value and show the relevant properties (1 in this case)	<code>['LEEDS', 'Terraced', 3, 245000]</code>	Program worked as expected

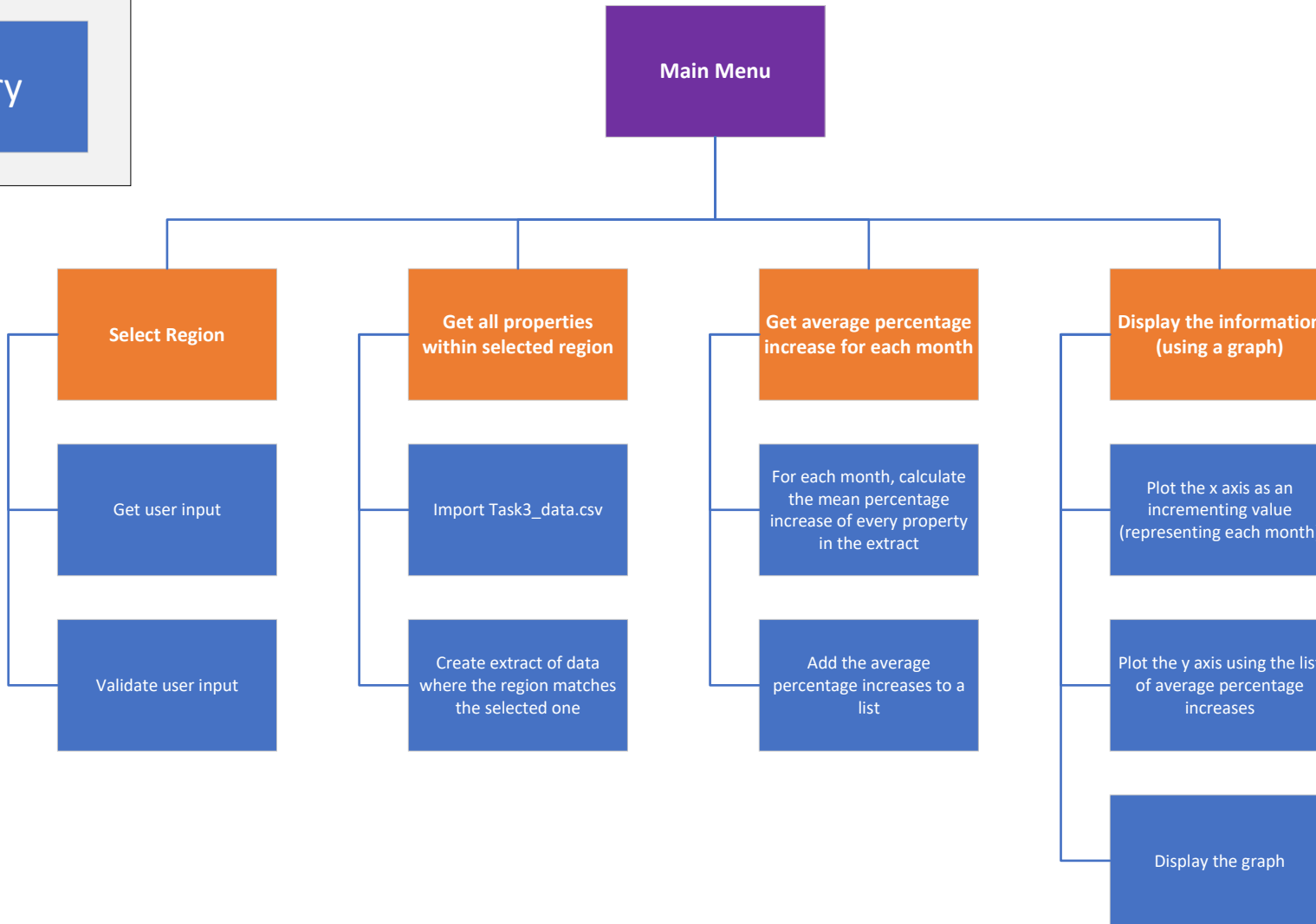
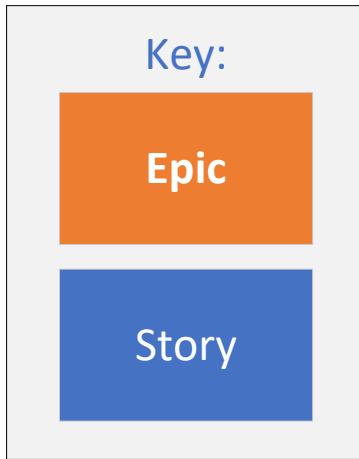
Testing Option 2: Entering an invalid region	Norwich	The program will give the user an error message before asking them to enter the input again	Please enter region: Norwich Entered region is not valid	Program worked as expected
Testing Option 2: Entering an invalid region including numbers	12345678	The program will give the user an error message before asking them to enter the input again	Please enter region: 12345678 Entered region is not valid	Program worked as expected
Testing Option 3: Entering a valid customer forename	customer_forename = █	The program will accept the value and move on	The program accepted the value and moved on	Program worked as expected
Testing Option 3: Leaving a blank customer forename	customer_forename =	The program will not accept the value and ask the user to enter the input again	Test 1: The program incorrectly accepted the value and moved on  Test 2: The program asked the user to enter the input again	Test 1: The program needs to only accept the input if the user entered a value <pre>customer_forename = "" while not customer_forename:     customer_forename = input('Please</pre> Test 2: Program worked as expected
Testing Option 3: Entering a valid customer surname	customer_surname = █	The program will accept the value and move on	The program accepted the value and moved on	Program worked as expected
Testing Option 3: Leaving a blank customer surname	customer_surname =	The program will not accept the value and ask the user to enter the input again	Test 1: The program incorrectly accepted the value and moved on  Test 2: The program asked the user to enter the input again	Test 1: The program needs to only accept the input if the user entered a value <pre>customer_surname = "" while not customer_surname:     customer_surname = input('Please</pre> Test 2: Program worked as expected
Testing Option 3: Are all the properties shown correctly?		The available properties should be: <pre>['LONDON', 'Terraced', 3, 735000] ['CARDIFF', 'Semi-Detached', 2, 100000] ['LEEDS', 'Terraced', 3, 245000] ['LONDON', 'Semi-Detached', 1, 240000]</pre>	<pre>['LONDON', 'Terraced', 3, 735000] ['CARDIFF', 'Semi-Detached', 2, 100000] ['LEEDS', 'Terraced', 3, 245000] ['LONDON', 'Semi-Detached', 1, 240000]</pre>	Program worked as expected
Testing Option 3: Selecting a valid option for a house to purchase	select = 1	The program should accept the value and move on	Test 1: The program gets stuck in this menu and there is no way to get out  Test 2: The program accepted the value and moved on	Test 1: The program tries to perform arithmetic operations on the input without casting it as an integer Fix: <pre>select = int(input(</pre> This won't cause any issues if the user enters a non-numeric character because the code is already in a try-except block
Testing Option 3: Selecting an option that is out of range for a house to purchase	select = 6	The program will give the user an error message before asking them to enter the input again	Test 1: The program handled the user error but did not give any error message  Test 2: <pre>Please select a purchase: 6 Please enter a number between 1 and 4</pre>	Test 1: I added an else statement in order to give the user an error message if the value they entered is out of range <pre>else:     print("Please enter a number between 1 and " + str(len(houses)))</pre> Test 2: Program worked as expected

<p>Testing Option 3: Selecting a non-numeric option for a house to purchase</p>	<p>select = option two</p>	<p>The program will give the user an error message before asking them to enter the input again</p>	<pre>Please select a purchase: option 2 ERROR PLEASE ENTER A VALID PROPERTY</pre>	<p>Program worked as expected</p>
<p>Testing Option 3: Does the receipt show the correct values? #1  (property value over £100000)</p>	<p>customer_forename = John customer_surname = Smith select = 1</p>	<p>Output should show:  Customer Receipt  FORENAME:John SURNAME: Smith PROPERTY COST: £735000 WITH STAMP DUTY: £750700.0  TRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES DATABASE  [ 'LONDON','Terraced', 3, 735000]</p>	<p>Test 1: Customer Receipt  FORENAME:John SURNAME: Smith PROPERTY COST: 735000 WITH STAMP DUTY: 865000.0 TRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES DATABASE  ['CARDIFF', 'Semi-Detached', 2, 100000]  Test 2: Customer Receipt  FORENAME:John SURNAME: Smith PROPERTY COST: £735000 WITH STAMP DUTY: £750700.0  TRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES DATABASE  ['LONDON', 'Terraced', 3, 735000]</p>	<p>Test 1: The program worked out the stamp duty incorrectly and the property that is deleted from the houses array is the wrong one. The monetary values are not formatted as currency. <pre>if sub_total &gt; 100000:     total_fees += 3000 + ((sub_total-100000) * 0.02) else:     total_fees += sub_total * 0.03</pre> I adjusted the values to make the program work out stamp duty correctly <pre>print (houses[select-1]) del houses[select-1]</pre> I subtracted 1 from select in this code to make it delete the correct property from the houses array <pre>sale.append(round(sub_total,2)) sale.append(round(final_total,2))</pre> This code round the totals to 2 decimal places (to fit with GBP) <pre>PROPERTY COST: £{} WITH STAMP DUTY: £{}</pre> I added "£" symbols when outputting the values so they are in a currency format  Test 2: Program worked as expected</p>
<p>Testing Option 3: Does the receipt show the correct values? #2  (property value up to £100000)</p>	<p>customer_forename = Amy customer_surname = Green select = 2</p>	<p>Output should show:  Customer Receipt  FORENAME:Amy SURNAME: Green PROPERTY COST: £100000 WITH STAMP DUTY: £103000.0  TRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES DATABASE  ['CARDIFF', 'Semi-Detached', 2, 100000]</p>	<p>Customer Receipt  FORENAME:Amy SURNAME: Green PROPERTY COST: £100000 WITH STAMP DUTY: £103000.0  TRANSACTION COMPLETE - PROPERTY REMOVED FROM SALES DATABASE  ['CARDIFF', 'Semi-Detached', 2, 100000]</p>	<p>Program worked as expected</p>
<p>Testing Option 3 and 5: Does the Show sales option show the sale that has just been made?</p>	<p>Inputs from recording the sale:  customer_forename = Amy customer_surname = Green select = 2</p>	<p>The sale data should show:  Forename Surname Property cost Total ['Amy', 'Green', 100000, 103000.0]</p>	<pre>Forename Surname Property cost Total ['Amy', 'Green', 100000, 103000.0]</pre>	<p>Program worked as expected</p>
<p>Testing Option 4:</p>		<p>The program should print the 4 available regions and ask the user for an input</p>	<pre>['LONDON', 'LEEDS', 'CARDIFF', 'BRISTOL'] Please enter the region of this property:</pre>	<p>Program worked as expected</p>

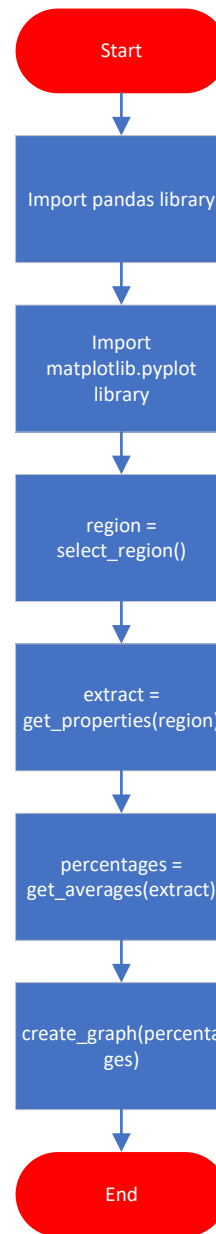
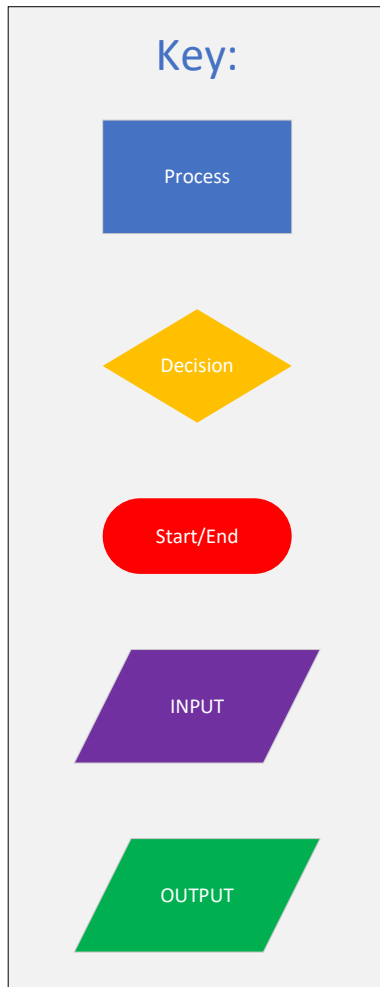
Does The program show the list of the company's supported regions and ask the user to enter the one for the new property?				
Testing Option 4: Entering a valid region	region = LONDON	The program should accept the value and move on	The program accepted the value and moved on	Program worked as expected
Testing Option 4: Entering a valid region in lowercase	region = london	The program should accept the value and move on	The program accepted the value and moved on	Program worked as expected
Testing Option 4: Entering an invalid region	region = NEWCASTLE	The program should provide the user with an error message before asking them to enter the input again	Please enter the region of this property: NEWCASTLE You can only enter one of the regions shown above	Program worked as expected
Testing Option 4: Does the program provide the user with a list of all the house types before asking them to enter the one for the new property?		The program should print the 3 available house types and ask the user for an input	House types: TERRACED, SEMI-DETACHED, DETACHED Please enter the house type: █	Program worked as expected
Testing Option 4: Entering a valid house type	house_type = TERRACED	The program should accept the value and move on	The program accepted the value and moved on	Program worked as expected
Testing Option 4: Entering a valid house type in lowercase	house_type = terraced	The program should accept the value and move on	The program accepted the value and moved on	Program worked as expected
Testing Option 4: Entering an invalid house type	house_type = mansion	The program should provide the user with an error message before asking them to enter the input again	Please enter the house type: mansion You can only enter one of the house types shown above	Program worked as expected
Testing Option 4: Entering a valid number of bedrooms	bedrooms = 3	The program should accept the value and move on	The program accepted the value and moved on	Program worked as expected
Testing Option 4: Entering an invalid number of bedrooms	bedrooms = -2	The program should provide the user with an error message before asking them to enter the input again	Please enter the number of bedrooms: -2 You can only enter a number (and it must be above 0)	Program worked as expected
Testing Option 4: Entering a non-numeric number of bedrooms	bedrooms = twelve	The program should provide the user with an error message before asking them to enter the input again	Please enter the number of bedrooms: twelve You can only enter a number (and it must be above 0)	Program worked as expected
Testing Option 4:	cost = 250000	The program should accept the value and move on		Program worked as expected

Entering a valid house cost				
Testing Option 4: Entering an invalid house cost	cost = 0	The program should provide the user with an error message before asking them to enter the input again	Please enter the house's cost: 0 You must enter the house's cost as a positive integer	Program worked as expected
Testing Option 4: Entering a non-numeric house cost	cost = lots of money	The program should provide the user with an error message before asking them to enter the input again	Please enter the house's cost: lots of money You must enter the house's cost as a positive integer	Program worked as expected
Testing Option 4: Does the program output the data the user has entered and give them the option to accept it or re-enter?	region = LONDON house_type = TERRACED bedrooms = 3 cost = 250000	The program should print the details the user entered which is: ['LONDON', 'TERRACED', 3, 250000] In the previous example. The program should then ask the user if this looks good.	['LONDON', 'TERRACED', 3, 250000] Does this look good? (Y/N)	Program worked as expected
Testing Option 4: Entering a valid choice for if the entry is correct #1	choice = YES	The program should accept the input and print a message showing the property has been added	Property added!	Program worked as expected
Testing Option 4: Entering a valid choice for if the entry is correct #2	choice = no	The program should accept the value and restart the process of entering the details for a new property	The program accepted the value and restarted the process of the user entering details for a new property	Program worked as expected
Testing Option 4: Entering an invalid choice for if the entry is correct	choice = 24	The program should not accept the value and prompt the user to enter the choice again	Does this look good? (Y/N) 24 Does this look good? (Y/N)	Program worked as expected
Testing Option 4: Does the new property show up in the houses list?	House details: region = LONDON house_type = TERRACED bedrooms = 3 cost = 250000 choice = yes  choice on menu afterwards: menu_selection = 1	After the user enters these details to add a new property, choosing option 1 on the menu should show that the new property is now part of the houses list	['LONDON', 'Terraced', 3, 735000] ['CARDIFF', 'Semi-Detached', 2, 100000] ['LEEDS', 'Terraced', 3, 245000] ['LONDON', 'Semi-Detached', 1, 240000] ['LONDON', 'TERRACED', 3, 250000]  The list shows the new property at the bottom with all the correct details	Program worked as expected

# Decomposition



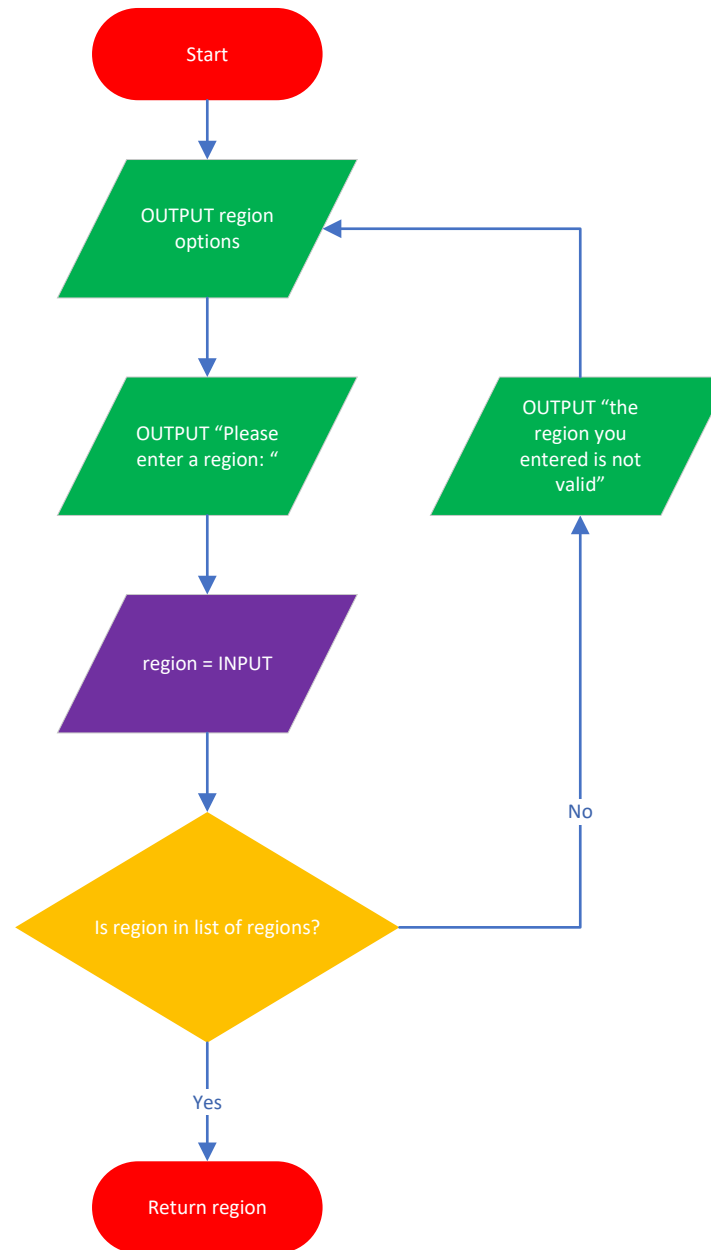
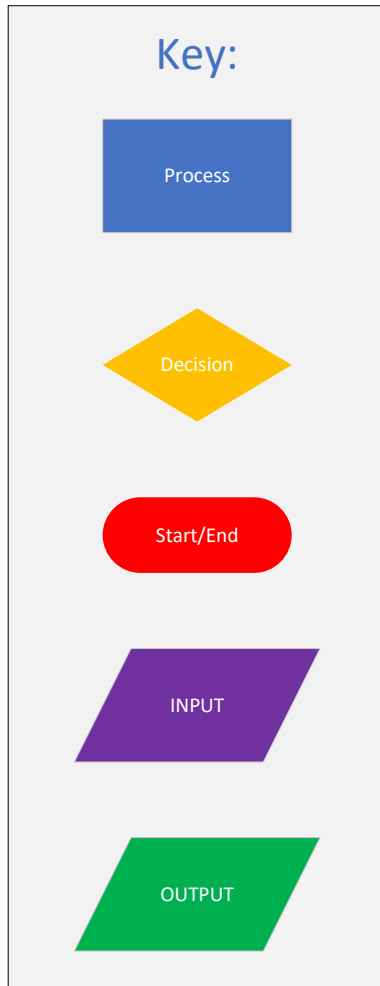
# Main Menu





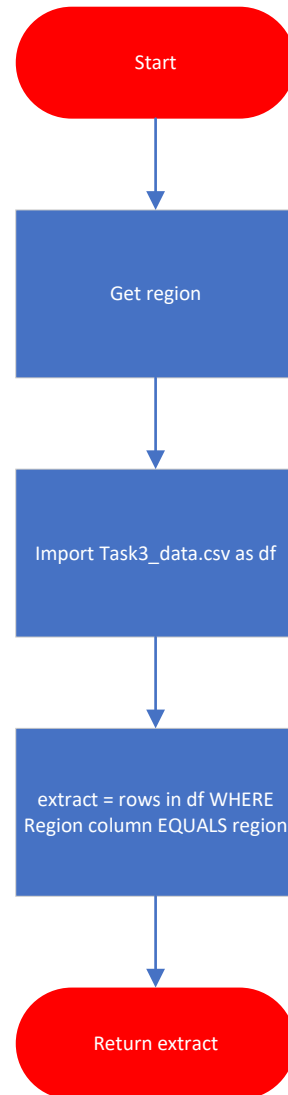
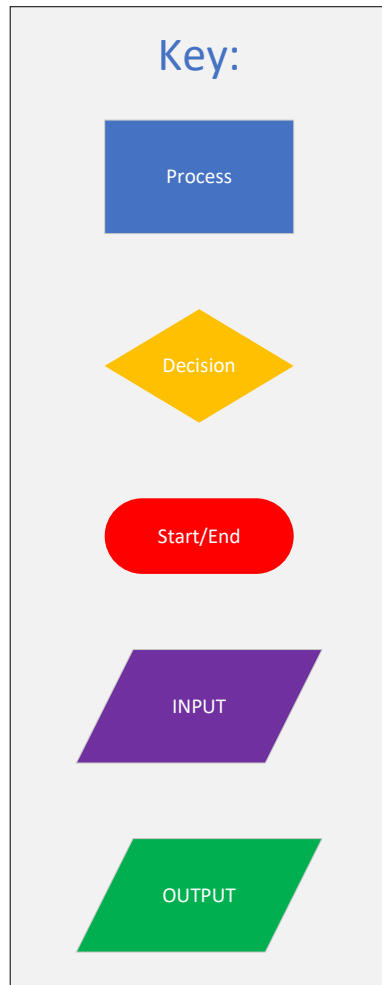
# Select Region

## select\_region()



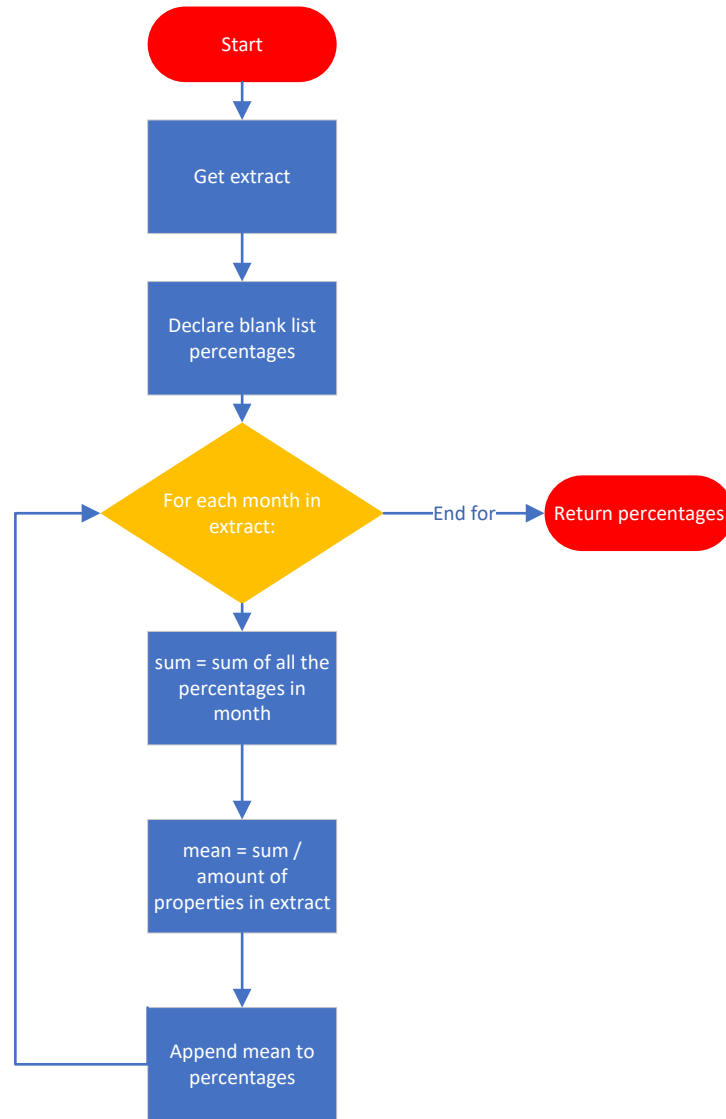
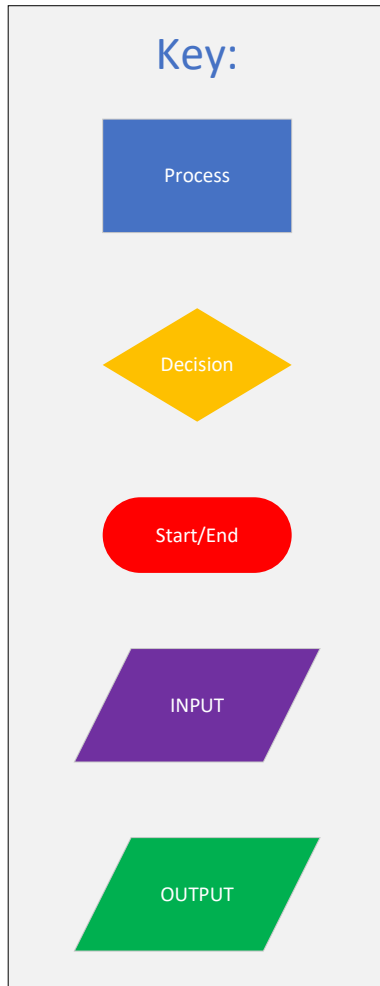
# Get all properties in selected region

## get\_properties(region)

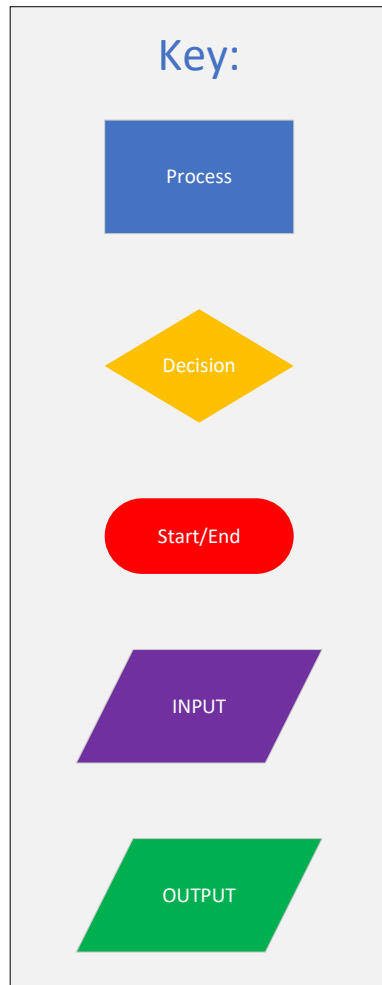


# Get average percentage increase for each month

## get\_averages(extract)



# Display the information create\_graph(percentages)



```

import datetime
import pandas as pd
import matplotlib.pyplot as plt

#reads the csv file into a pandas dataframe
df = pd.read_csv('Task4a_data.csv')

#creates a dictionary which can be used to link a region code to a region
region_dict = {
    "1": "London",
    "2": "Bristol",
    "3": "Cardiff",
    "4": "Leeds",
    "5": "Manchester",
    "6": "Birmingham",
    "7": "Edinburgh",
    "8": "Glasgow"
}

#Gives the user a menu of options to choose between and will return their
choice
def mainmenu():
    print("\t\t****Welcome to the Dashboard****")
    print('1) Return all current data')
    print('2) Return data for a specific region')
    print('3) Return data for different property types within a specific
region')
    print('4) Return data for different property sizes within a specific
region')
    print('5) Compare overall increase in property value by region')
    print('To quit, enter any other value')
    #I removed the int() from the input in order to prevent the program
from crashing when a non-numeric value is entered
    return input("")

#prints all the data from the csv file
def alldata():
    print(df)

#displays a line graph showing the average percentage increase for a
specified region over a specified time period
def region_check(region, startdate, enddate): # region, startdate,
enddate
    #creates an extract of the dataframe which only has the data within
the specified time period
    df1 = df.loc[:, startdate:enddate]
    df2 = df.loc[:, 'Region Code':'Rooms']

    result = pd.concat([df2, df1], axis=1,
join='inner').where(df2["Region"] == region)
    result = pd.DataFrame(result)
    result.dropna(inplace=True)
    print(result)
    #plots a line graph using the mean percentage increase from each
month within the specified time period
    ave = df1.mean()
    ave.plot()

```

```

    #titles the graph appropriatley and includes the values inputted by
the user
    plt.title("Average percentage increase in " + region + " from " +
startdate + " to " + enddate)
    plt.show()
    return result

#displays a line graph showing the average percentage increase for each
property type in a specified region over a specified time period
def property_types(region, startdate, enddate):
    #creates an extract of the dataframe only including rows where the
region matches the "region" value and only includes the columns that are
between the start and end date
    df1 = df.loc[(df["Region"] == region)].loc[:, startdate:enddate]
    #this line will use the concat function to add back in the "Property
Type" column to the dataframe extract
    df1 = pd.concat([df["Property Type"], df1], axis=1, join='inner')

    for property_type in df1["Property Type"].unique():
        extract = df1.loc[(df1["Property Type"] == property_type)]
        #this line removes the "Property Type" column again as to not
include it when determining the mean of all the values
        extract = extract.iloc[:, 1:]
        #plots a line of the average percentage increase for this
property type in this region over time
        extract.mean().plot()
        plt.legend([property_type for property_type in df1["Property
Type"].unique()], title="Property Types:")
        #titles the graph appropriatley and includes the values inputted by
the user
        plt.title("Different property types in " + region + " from " +
startdate + " to " + enddate)
        #displays the graph
        plt.show()

#displays a line graph showing the average percentage increase for each
property size in a specified region over a specified time period
def property_sizes(region, startdate, enddate):
    #creates an extract of the dataframe only including rows where the
region matches the "region" value and only includes the columns that are
between the start and end date
    df1 = df.loc[(df["Region"] == region)].loc[:, startdate:enddate]
    #this line will use the concat function to add back in the "Rooms"
column to the dataframe extract
    df1 = pd.concat([df["Rooms"], df1], axis=1, join='inner')

    for room_count in df1["Rooms"].unique():
        extract = df1.loc[(df1["Rooms"] == room_count)]
        #this line removes the "Rooms" column again as to not include it
when determining the mean of all the values
        extract = extract.iloc[:, 1:]
        #plots a line of the average percentage increase for this
property type in this region over time
        extract.mean().plot()
        plt.legend([room_count for room_count in df1["Rooms"].unique()],
title="Rooms:")
        #titles the graph appropriatley and includes the values inputted by
the user

```

```

plt.title("Different sized properties in " + region + " from " +
startdate + " to " + enddate)
#displays the graph
plt.show()

#displays a bar graph showing the difference between the overall increase
in property value from different regions over a specified time period
def compare_overall_increase(startdate, enddate):
    #this array will hold the overall increase of property value for each
region
    totals = []
    #will iterate through each region present in the dataframe
    for region in df["Region"].unique():
        #creates an extract of the dataframe only including rows where
the region matches the "region" value and only includes the columns that
are between the start and end date
        df1 = df.loc[(df["Region"] == region)].loc[:, startdate:enddate]
        #gets the sum of all percentages in the extract, round it to 2
decimal places and adds it to the totals array
        total_increase = round(df1.sum("columns").sum(), 2)
        totals.append(total_increase)

    #plots a bar graph with the region names on the x axis and the
corrospounding totals as the y axis
    #this will show in a visually meaningful way, the difference between
the overall increase in property value from different regions
    plt.bar([region for region in df["Region"].unique()], totals)
    #titles the graph appropriatley and includes the values inputted by
the user
    plt.title("Overall increase in property value from " + startdate + "
to " + enddate)
    #displays the graph
    plt.show()

#gives the user a list of all the regions and asks them to pick one, the
input is validated and then returned
#this code was moved to be in a function so that its functionality can be
used throughout the program rather than needing to add unnecessary lines
of code
def get_region():
    while True:
        #presents the user with an options menu of all the regions
present in the dataframe
        print("Regions:")
        for i, region in enumerate(df["Region"].unique()):
            print(str(i+1) + ")", region)

        #allows the user to enter the region as a code from 1 to 8
        region_code = input("Please enter the name of the region you
would like to check: ")
        #validates that the input is a number between 1 and 8
        if region_code.isnumeric() and 0 < int(region_code) < 9:
            #converts the code into the region and makes sure it is
within the dataframe
            region = region_dict[region_code]
            if region in df.Region.values:
                return region
            else:

```

```

        print("Region not found")

#will ask the user for a start and end date, validate them and return the
values
#this code was moved to be in a function so that its functionality can be
used throughout the program rather than needing to add unnecessary lines
of code
def get_dates():
    while True:
        startdate = input("PLEASE ENTER A START DATE AS MONTH-YEAR e.g.
JAN-20: ")
        startdate = startdate.capitalize()
        #determines if the entered date is valid by checking to see if it
is within the dataframe
        if startdate not in df.columns:
            print("Error start date not found")
        else:
            while True:
                enddate = input("PLEASE ENTER AN END DATE AS MONTH-YEAR
e.g. JAN-20: ")
                enddate = enddate.capitalize()
                #determines if the entered date is valid by checking to
see if it is within the dataframe
                if enddate not in df.columns:
                    print("Error end date not found")
                else:
                    return [startdate, enddate]

#main loop
x = mainmenu()
#will keep looping the menu until the user enter a value that isn't
within the menu of options
while x in ["1", "2", "3", "4", "5"]:
    if x == "1":
        alldata()

        #asks the user for a region, startdate and enddate, makes sure they
are valid and calls the region_check function using them as the
parameters
        elif x == "2":
            region = get_region()
            startdate, enddate = get_dates()
            region_check(region, startdate, enddate)

        #asks the user for a region, startdate and enddate, makes sure they
are valid and calls the property_types function using them as the
parameters
        elif x == "3":
            region = get_region()
            startdate, enddate = get_dates()
            property_types(region, startdate, enddate)

```



```
#asks the user for a region, startdate and enddate, makes sure they
are valid and calls the property_sizes function using them as the
parameters
```

```
elif x == "4":
    region = get_region()
    startdate, enddate = get_dates()
    property_sizes(region, startdate, enddate)
```

```
#asks the user for startdate and enddate, makes sure they are valid
and calls the compare_overall_increase function using them as the
parameters
```

```
elif x == "5":
    startdate, enddate = get_dates()
    compare_overall_increase(startdate, enddate)
```

```
#calls the menu again to ask the user for another choice
x = mainmenu()
```

# Task4b Review

In this evaluation I will outline how the solution produced meets the system and user requirements as well as suggesting how the solution could be further developed.

## Main Menu

The main menu provides the user with a list of options and will run the corresponding functions based on their input.

```
def mainmenu():
    print("\t\t****Welcome to the Dashboard****")
    print('1) Return all current data')
    print('2) Return data for a specific region')
    print('3) Return data for different property types within a specific
region')
    print('4) Return data for different property sizes within a specific
region')
    print('5) Compare overall increase in property value by region')
    print('To quit, enter any other value')
```

This menu displays the choices to the user in a meaningful way and is easy to use as the user only needs to enter a number for each option. The code can handle user errors such as entering a number out of range or not entering a number at all.

The main loop will keep running the menu until the user chooses to quit by entering any value that isn't one of the menu options.

## Get\_region function

This function will present the user with a list of all the regions present in the data and like the main menu, ask them to enter a number corresponding to one of the regions. If the user enters an incorrect value, the code will handle the error and ask them again for the input.

```
print("Regions:")
for i, region in enumerate(df["Region"].unique()):
    print(str(i+1) + ")", region)
```

*The above code will print all the unique regions present in the data frame and put a number in the front to show which number the user needs to enter to choose that region*

```
    #allows the user to enter the region as a code from 1 to 8
    region_code = input("Please enter the name of the region you
would like to check: ")
```

The functionality of this code is easy to use because it only requires the user to enter a single digit and because it displays the options in a clear and meaningful way.

## Get\_dates function

This function will ask the user for a start and end date, validate them and return the values. This code is easy to use as it clearly displays what the user needs to enter by giving them an example of what they need to enter.

```
    startdate = input("PLEASE ENTER A START DATE AS MONTH-YEAR e.g.
JAN-20: ")
```

The code will check the data frame to see if the user's input is one of the column names. If the input was not present in the data frame, the user will be prompted to enter the date again. The same process is repeated for getting the "enddate" and the values are returned once they are both valid.

## Return data for a specific region

This section of the code is designed to meet the first of the system requirements which is to allow users to select a region and show the trends for increases in property over time.

```
#asks the user for a region, startdate and enddate, makes sure they
are valid and calls the region_check function using them as the
parameters
elif x == "2":
    region = get_region()
    startdate, enddate = get_dates()
    region_check(region, startdate, enddate)
```

This code will run when the user chooses the second option on the main menu and it will first call the "[get\\_region](#)" function and retrieve the "region" variable.

It will then get the "startdate" and "enddate" variables from calling the "[get\\_dates](#)" function.

Finally, the code will call the "region\_check" function with these values as parameters.

### Region\_check function

This function will take the "region", "startdate" and "enddate" variables and produce a line graph showing average percentage increase of properties in that region between the two time periods.

```
df1 = df.loc[:, startdate:enddate]
df2 = df.loc[:, 'Region Code':'Rooms']

result = pd.concat([df2, df1], axis=1,
join='inner').where(df2["Region"] == region)
result = pd.DataFrame(result)
result.dropna(inplace=True)
```

*The above code creates an extract of the data frame which only has the data within the specified time period. And will then filter it to only show records relating to the selected region.*

```
ave = df1.mean()
ave.plot()

plt.title("Average percentage increase in " + region + " from " +
startdate + " to " + enddate)
plt.show()
return result
```

*The above code will plot a graph showing the mean percentage increase from each month within the specified time period, give the graph a title that is meaningful to the user and display the graph.*

The information being displayed in a graph makes use of appropriate graphical output in way that is relevant to the end user, the information is clear to understand and is displayed in a meaningful way.

## Return data for different property types within a specific region

This section of the code is designed to meet part of the system requirement for the code to identify trends and patterns over time for specific property types of different sizes within a selected region. I

decided to break this requirement down into two functions, one for property types (this one) and another for property sizes.

```
#asks the user for a region, startdate and enddate, makes sure they
are valid and calls the property_types function using them as the
parameters
elif x == "3":
    region = get_region()
    startdate, enddate = get_dates()
    property_types(region, startdate, enddate)
```

This code will run when the user chooses the third option on the main menu and it will first call the [“get\\_region”](#) function and retrieve the “region” variable.

It will then get the “startdate” and “enddate” variables from calling the [“get\\_dates”](#) function.

Finally, the code will call the “property\_types” function with these values as parameters.

### Property\_types function

This function will display a line graph showing the average percentage increase for each property type in a specified region over a specified time period.

```
extract.mean().plot()
plt.legend([property_type for property_type in df1["Property
Type"].unique()], title="Property Types:")
#titles the graph appropriately and includes the values inputted by
the user
plt.title("Different property types in " + region + " from " +
startdate + " to " + enddate)
#displays the graph
plt.show()
```

*This code plots and displays the graph*

The graph will show a different line for each different property type, this displays the information to the user in a clear and meaningful way and allows them to easily compare the differences between the property types. The graph has a key showing which line is which making it easy to read and a title that tells the user in a meaningful way what the graph shows.

### Return data for different property sizes within a specific region

This section of the code is designed to meet part of the system requirement for the code to identify trends and patterns over time for specific property types of different sizes within a selected region. I decided to break this requirement down into two functions, one for property types and another for property sizes (this one).

```
#asks the user for a region, startdate and enddate, makes sure they
are valid and calls the property_sizes function using them as the
parameters
elif x == "4":
    region = get_region()
    startdate, enddate = get_dates()
    property_sizes(region, startdate, enddate)
```

This code will run when the user chooses the third option on the main menu and it will first call the [“get\\_region”](#) function and retrieve the “region” variable.

It will then get the “startdate” and “enddate” variables from calling the [“get\\_dates”](#) function.

Finally, the code will call the “property\_sizes” function with these values as parameters.

## Property\_sizes function

This function will display a line graph showing the average percentage increase for each property size in a specified region over a specified time period.

```
        extract.mean().plot()
        plt.legend([room_count for room_count in dfl["Rooms"].unique()],
title="Rooms:")
        #titles the graph appropriately and includes the values inputted by
the user

        plt.title("Different sized properties in " + region + " from " +
startdate + " to " + enddate)
        #displays the graph
        plt.show()
```

*This code plots and displays the graph*

The graph will show a different line for each different property size, this displays the information to the user in a clear and meaningful way and allows them to easily compare the differences between the property sizes. The graph has a key showing which line is which making it easy to read and a title that tells the user in a meaningful way what the graph shows.

## Compare overall increase in property value by region

This section of code is designed to meet the system requirement to identify trends and patterns over time for the region with the highest overall increase in property value. I decided to display this information using a bar graph in order to show the user in a clear and meaningful way, the comparison between all the regions within the specified time period, with the region with the most overall percentage increase being obvious because of it having the biggest bar.

```
        #asks the user for startdate and enddate, makes sure they are valid
and calls the compare_overall_increase function using them as the
parameters
        elif x == "5":
            startdate, enddate = get_dates()
            compare_overall_increase(startdate, enddate)
```

First the code will get the "startdate" and "enddate" variables from calling the "[get\\_dates](#)" function.

Then it will call the "compare\_overall\_increase" function with these dates as parameters.

## Compare\_overall\_increase function

This function takes a "startdate" and "enddate" value and creates an extract of the data frame using only the dates within this time frame. It will then work out the overall average percentage increase for every region and display that information in a bar graph.

```
        for region in df["Region"].unique():
            #creates an extract of the dataframe only including rows where
the region matches the "region" value and only includes the columns that
are between the start and end date
            dfl = df.loc[(df["Region"] == region)].loc[:, startdate:enddate]
            #gets the sum of all percentages in the extract, round it to 2
decimal places and adds it to the totals array
            total_increase = round(dfl.sum("columns").sum(), 2)
            totals.append(total_increase)
```

*This code creates the extract of the data frame only including correct dates and works out the total percentage increase for each region.*

```
#plots a bar graph with the region names on the x axis and the
corresponding totals as the y axis
#this will show in a visually meaningful way, the difference between
the overall increase in property value from different regions
plt.bar([region for region in df["Region"].unique()], totals)
#titles the graph appropriately and includes the values inputted by
the user
plt.title("Overall increase in property value from " + startdate + "
to " + enddate)
#displays the graph
plt.show()
```

*The above code will plot and display the bar graph*

## What could be improved

Even though the code I wrote uses accepted conventions for variable and function names consistently, the pre-existing code contains some names that don't use accepted naming conventions such as "mainmenu()", "alldata()", "startdate" and "enddate". The code could be corrected to change these names to make the code slightly more readable/maintainable.

The code could also be improved by having a more sophisticated graphical user interface throughout the program including the main menu. This would provide the user with an even better experience and make the solution easier to use as well as display the information in a way that is clear to understand and more visually meaningful to the end user.

## Conclusion

Overall, the solution I have developed meets all of the system requirements as well as carefully considering the experience of the end user. I have chosen to display the information using line and bar graphs in order to make the information easily understood and visually meaningful to the user. My solution is also easy to use by giving the user clear instructions of how to use the program and what inputs they need to enter.

## Mark Grid

<b>Task</b>	<b>Assessment Focus</b>	<b>Learner score</b>	<b>Mark Scheme Band</b>	<b>Maximum Mark</b>
<b>1: Planning a project</b>	Gantt chart	5	3	6
	Resource and cost plan	3	2	4
	Rationale	3	1	9
<b>2: Identifying and fixing defects in existing code</b>	Use of testing to identify defects	6	3	8
	Understanding of the testing process	4	3	4
	The solution	9	3	9
<b>3: Designing a solution</b>	Decomposition of the problem	8	3	8
	Application of logical thinking and conventions	5	3	6
	Communication of the design	3	3	3
<b>4a: Developing the solution</b>	Functionality	5	3	6
	Logic and programming structures	3	3	3
	Robustness	2	2	3
	Security	6	3	6
	Code organisation	8	4	8
	User experience	7	4	8
<b>4b: Reflective evaluation</b>	Programming outcomes	3	2	6
	Future Developments	2	2	3
<b>Total Mark</b>		<b>82</b>		<b>100</b>





Explore Pearson's  
T Levels offering at  
[quals.pearson.com/tlevels](https://quals.pearson.com/tlevels)

Copyright in this document belongs to, and is used under licence from, the Institute for Apprenticeships and Technical Education, © 2023. 'T-LEVELS' is a registered trade mark of the Department for Education. 'T Level' is a registered trade mark of the Institute for Apprenticeships and Technical Education. 'Institute for Apprenticeships & Technical Education' and logo are registered trade marks of the Institute for Apprenticeships and Technical Education.

The T Level Technical Qualification in Digital Production, Design and Development is a qualification approved and managed by the Institute for Apprenticeships and Technical Education.

Pearson Education Limited is authorised by the Institute for Apprenticeships and Technical Education to develop and deliver this Technical Qualification.

Pearson and logo are registered trade marks of Pearson.