1,600 words  
  
Required:  
coding techniques  
importance of testing  
TDD – test as we go

Error log and ways to fix  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
**Intro:**  
  
*How application work listed in githuib, will analyse the initial design, how scope changed and lessons learnt for fufuture development, each part of the program will be al;anysed in the logical order of development (aligned with git branch order, numbered)  
 In addition to   
  
The content of this report will work through the xdevelopment process entailed to creatre the python program for assessment, it will cover the planning, implementation, refactoring and testing stages of the project.  
  
python packages, good but also too extracted or presume ok, may miss boundary conditions.  
Also costs added like fluent assertions. Main python can have deprecated features   
  
Python syntax, screaming snake case but Ide used snake case – could use linter but IDE ok  
Functions need colon, sometimes too simple to see what’s in and out of function or what logic block it’s inside.*  
  
The objective of this report is to analyse the development process ofverall, assess what improvements could be made with the experience and knowledge gained from completing this assignment. Additionally each element of the developed program will be looked at specifically to see it’s evolution, how it was tested and where design decisions were taken.  
  
   
  
  
Lack of AI  
  
**Planning**  
two parts, initial well planned (demo) then rapid scope increase and the issue of working like that. Mention git  
Key ssumptions (onenote )  
Using CSV for ease  
  
5 planes of user experience by

Garrett, Jesse James. (2011) The Elements of User Experience. Berkeley, CA. New Riders

Luca’s focus to look at what is important and less what it looks like. Minimum clicks to complete task and look at sites involving money, must be super clear.  
  
Screenshot of planning, ERD/ unified modelling language not all fields used  
  
  
  
**Implementing  
  
Started with GIT repo  
  
Using CSV**  
Exhaustion  
python ease, material online, packages.  
Did not implement sql db as wanted to complete MVP, and following YAGNI for simplicity, however later realise too fundamental of a feature to not include for things like deletion, upload etc.  
  
  
  
**talk through each part.**  
  
Initial results generation numpy – testing throughout  
  
Lots of screenshots in onenote – mean average calculation vs manual  
Pass/fail count etc.  
  
AI got it wrong  
  
  
Screenshot of curtailed print in ‘design cecisions’ onenote  
  
**Tkinter GUI – layout**  
  
Search resul;ts  
  
Left hand results show (would change that to table later)  
  
Student result right  
  
API with image generation  
  
Graphs  
  
Refactoring issues  
Tightly coupled, plan better but also is it possible to make code not as tightly coupled ? Ciorcular steps that call one another.  
  
Unit tests  
  
  
**Additional scope**  
  
Sepertate files  
  
Sql lite DB – tutoria[l link](https://www.youtube.com/watch?v=pd-0G0MigUA)   
More graphs, data analysis