You will submit your code files electronically to the project folder on VITAL. You will also produce a 2-4 page report to accompany your program which will explain its functionality in detail. Your main script, function files, any data files and report should be combined into a zip file and uploaded together. The zip folder should be named using your student ID number. More detailed instructions on the submission process will be included in the project folder.

Project Requirements:

You are to work individually to create a project and the solution. This means that YOU are responsible for writing the code and report. The project must be a menu-driven, modular program that includes at a minimum:

- User interaction (with the menu and input functions)
- Reading from at least one data file.
- At least one function that has one or more input arguments.
- At least one function that has one or more output arguments – this can be the same as the function with input arguments above.
- At least one function with no input arguments.
- At least one function with no output arguments.
- Output (both plotting and fprintf).
- Some calculations/statistical analysis.
- Use of vectors and matrices, including indexing.
- At least one selection statement (if, if-else, switch, etc.)
- At least one loop statement (for, while)

In the report you must explain where in your code you have each of these items. We have covered the first 6 chapters of the book (as well as sections 11.1-11.4 on Advanced Plotting) and so you will be expected to use this project to demonstrate your understanding of this material. In order to achieve high marks, you will have to go above and beyond the basics. There is a lot more that MATLAB is capable of than we have covered, go and explore a bit.

You can create your own data file(s) if you like, or you can look online for existing data files. Many government websites have data that can be downloaded. Some examples:

UK government collected data: https://data.gov.uk/ (this is a good place to start)
Weather: https://www.metoffice.gov.uk/research/climate/maps-and-data/historic-station-data
Solar: https://rredc.nrel.gov/solar/old_data/nsrdb/

Marking Criteria

	Outstanding	Comprehensive/ Competent	Deficient/Weak	*0*
CODE 50%	All functions present and properly named. Code runs perfectly with no errors. Exceeds minimum project requirement list.	All functions present. Code runs, but may have some small errors. Meets minimum project requirements.	Functions missing or improperly named. Code runs but has many errors. Missing project requirements	Code does not run or is missing
STYLE 25%	The best code that could be reasonably expected. Fully documented and demonstrates a clear understanding of MATLAB programming	Code is not fully documented. Some weaknesses or lack of understanding evident.	Code is very difficult to read or is otherwise confusing. Poorly documented.	Code is not documented at all.
REPORT 25%	Clearly and concisely explains the MATLAB code and how each part runs and contributes to the overall program. Well written, without spelling or grammatical mistakes.	MATLAB code is explained, but with some omissions. Report is somewhat unclear or contains some spelling/ grammatical mistakes.	MATLAB code is not well explained or report is difficult to read. The report does not demonstrate a clear understanding on the part of the authors. Major spelling/grammatical mistakes. **Report exceeds the 4 page limit.**	Report impossible to read or missing.

Pay particular attention to these key things from the above criteria:

- If the code does not run (you mis-named something, or forgot a file, or whatever), you will not get any marks for **CODE**.
- Markers will not troubleshoot your code. Either it works or it doesn't!
- Don't forget to include the data files your code reads from.
- If the code is not documented, meaning it does not have any comments to explain what it is doing, you will not get any marks for **STYLE.**
- If you exceed the 4 page limit for the report, you will get at most 1/5 marks for **REPORT**.