COMP1010 Major Project

Due Dates

Proposal: 11:59PM Sunday 18th of July (end of week 7)

Final prototype: 11:59PM Sunday 8th of August (end of week 10)

Marks

Proposal: 10 marks

Final prototype: 20 marks

Total: 30 marks

Overview

For this project, you will be designing and implementing a prototype web application to solve a problem of your own choice. At a minimum, this prototype will need to demonstrate that your solution to the problem works and is usable by the target users.

If you're having trouble thinking of a problem you wish to solve, a list of possible ideas is given below for inspiration.

Whatever you intend to do, you need to get your tutor to approve the idea before you submit your proposal. It is strongly advised to do this by week 6 (i.e., during your week 5 class) so you have sufficient time to take into account any feedback.

Proposal (due end of week 7)

Your proposal should be written in a Python Notebook. A template is provided here . Replace the sections in *italics* with the corresponding information about your own project. If you'd rather not use Collaboratory to edit this template, you can download it and edit it in VSCode or similar.

Please note that your proposal does not need to be of a specific length or word count, but you should ensure it contains all relevant information. Unless otherwise specified in the template, assume the reader has the same level of technical knowledge as another student in this course.

Demonstration of Feasibility

As part of your proposal, you will be writing code to demonstrate that a feature of your prototype is feasible to implement in Python.

What feature you implement (or partially implement) depends on your project, but as a general guide, it should be one of the ways in which your program either retrieves, processes, or displays information. Some examples are given in the list below.

Final Prototype (due end of week 10)

Your final prototype will be your best attempt at implementing what you proposed. In doing this you should keep in mind the following:

- Most of the marks available are for having a basic working version of your application (see marking scheme below). Because of this, your first focus should be on delivering your MVP (see proposal template).
- Projects like this often take longer than expected (even for experienced programmers), so you should ensure you start early and have sufficient time to solve any unexpected problems that may arise.

Your final prototype should include (either as a comment at the top of your .py file, or in a separate README.md file) the following table, filled out:

Category	Feature	Status

- Category: This should be either "MVP", "Nice to have" or "Stretch goal".
- Feature: This is the description of the feature as described in your proposal.
- Status: This should be either empty (for not attempted), "attempted" or "completed".

The features should appear in the same order as in your proposal. You may include additional features which were not included in your proposal. This table is not worth any marks, it is simply to assist your tutor in marking your assignment.

Assistance and Showing Progress

Some of the tutorial and lab time in weeks 8 and 9 will be dedicated to providing assistance with your project. You can use this as an opportunity to ask any questions you might have specifically about your project, as well as bounce ideas off the course staff.

Lab marks in weeks 8 and 9 will be awarded for demonstrating progress with your project. There are no specific criteria you will have to meet to get these marks, but you will need to show you have made progress compared to your work in the previous week.

Example ideas

Below is a list of possible project ideas. If you are looking for an idea, you can use one of the ones below as either inspiration or as your whole project. You are encouraged, but not required, to add your own spin on these ideas to make them your own.

- **Assignment Partner Finder**
  A common problem faced by students is finding partners for pair assignments. After COVID, university teaching primarily moved online. This has made it even harder to find others to work with. It would be useful for students to have an application that let them post that they're looking for a partner, alongside relevant information about themselves, and search for other students that "match" what they're looking for. There are many ways such an application could work. A suitable demonstration of feasibility might be showing how information regarding potential partners is presented to the user (most likely programmatically generated HTML).
- **WikiGuesser**
  A fun and interesting way to check your general knowledge is to be presented with a quote from a Wikipedia article then try to guess what article it is from. However, this is a hard thing to do by yourself as you need someone to give you the quote. An application would solve this problem as it can automatically pick out sentences from Wikipedia articles and present them to you. A suitable demonstration of feasibility might be code that picks a Wikipedia article at random and extracts a sentence from it.
- **Meme Generator**
  Memes have become one of the dominant forms of entertainment globally. A classic form of meme is a picture with text on the top and/or bottom. A basic web application for generating memes would let the user supply a URL for an image as well as some text they want written on it. A more advanced application could offer features like a standard library of images, different fonts, layouts, etc. A suitable demonstration of feasibility might be code that generates a single meme.
- **What 3 Australian Animals (hard)**
  Like what 3 words (https://what3words.com), but instead of dividing the whole world up into 3mx3m squares and giving them unique 3-word names, divide the UNSW campus up into 3mx3m squares and give each square a unique name consisting of 3 Australian animals. Even a basic version of an application for this would need to display locations on the UNSW map , as well as convert from 3-animal-names to map coordinates and vice versa.

Submission

You can submit your proposal and your final prototype via the course website at any time.

Proposal

For the proposal, submit your Python Notebook (the file should have an ‘. ipynb' extension). If you're using Collaboratory, you can get this by clicking **File** then **Download .ipynb** .

In the event you submit late, a late penalty will be applied to your mark. That penalty is 2 marks per day off the maximum obtainable mark for up to 3 days.

Final prototype

If your prototype is just a single Python file (with a '.py' extension) you can submit that. If it is split over multiple files, create a zip file containing all of them and submit that.

In the event you submit late, a late penalty will be applied to your mark. That penalty is 2 marks per day off the maximum obtainable mark for up to 3 days.

Plagiarism

While you are encouraged to help, and get feedback from, other students in the course, the code you submit must be your own work or the work of your partner. Submission of code partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined both automatically and manually for such submissions.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

Do not provide or show your code to any other person, except for the teaching staff of COMP1010. If you knowingly provide or show your code to another person for any reason, and code derived from it is submitted, you may be penalised, even if the work was submitted without your knowledge or consent. This may apply even if your work is submitted by a third party unknown to you.

Note, you will not be penalised if your work has the potential to be taken without your consent or knowledge.

Marking Guidelines

For the project proposal, see the criteria in the template.

For the final prototype:

Criteria	Guideline	Mark
Stability (5 marks)	• The submitted code isn't valid Python and can't be executed (0-1 marks) • The submitted code has parts that are able to execute, but overall it can't be used without crashing (2 marks) • Mostly works, but crashes frequently (3 marks) • Works, but will crash under specific circumstances (4 marks) • No crashes or major failures noticed when run (5 marks)	•
Functionality (5 marks)	• Lacks any real functionality (0-1 marks) • Offers some functionality, but not enough to be useful to any user (2 marks) • Working MVP (3 marks) • Contains some "nice to have" features (4 marks) • An ambitious "stretch goal" has been met (5 marks)	•

Programming (5 marks)	• Code is not in Python or application consists entirely of static HTML pages (0-1 marks) • Code almost entirely consists of snippets copied from elsewhere (1-2 marks) • Code does not use appropriate control or data structures (e.g. using lists when a dictionary would be more appropriate) (3-4 marks) • Appropriate use of Python control structures, data structures, functions, and libraries (5 marks)	•
Interactivity (5 marks)	• Interface is indecipherable and/or multiple usability heuristics have been violated (0-1 marks) • Interface is usable, but not without looking at the code and/or some other resource to figure out how (2-3 marks) • Interface is usable, but some usability heuristics have been violated (4 marks) • Interface does not violate any usability heuristics and is generally easy to use (5 marks)	•

Note: The above table is only a *guideline.* Owing to the open-endedness of this project, how marks are awarded may vary slightly from what is above. In particular, full marks are unlikely to be awarded for unambitious projects.

Note: If you submit an application which is not a web application, marks will be capped at 50% if it does not have a graphical user interface.

Note: Up to 2 marks may be deducted off your total marks for not following good code style.