**Create a Development Environment Exercise**

Alan L. Dennis

University of the Cumberlands

**Create a Development Environment Exercise**

It is common when doing software development related to artificial intelligence to create or provision a development environment. Completing this project will enable you to complete the other projects in this course. This document outlines the instructions for the project. You will be presented with options for the project. Choose the one that best suits your hardware equipment and technology familiarity. Lastly, a conclusion will restate what was covered.

# General Instructions

After reviewing the lecture, use this document to create a development environment. Ensure the environment is linked to your GitHub Repository for this course. Create a simple application (with a name like Hello World). Add the application that you are using in your GetHub Repo. Document the process in a report. The report should be **two to three pages** of content and should include a title and references page using APA format. Document any challenges you met and mitigation strategies you utilized. Ensure that you document your progress with screenshots and a narrative. Do not simply paste images. Instead, use figures with names that contain screenshots and discuss them in the narrative.

# Anaconda and Jupyter Lab/Notebooks

Anaconda is a software package that makes it easy to install Python and related technologies. It simplifies the process of setting up an environment for data science, artificial intelligence, or machine learning (Anaconda, 2018). Once you have installed Anaconda, you will want to use Jupyter Notebooks. Jupyter Notebooks allow you to combine code, data, comments and results into an interactive notebook. There are multiple ways to start Jupyter Notebook; a simple one is to use the Anaconda prompt, as shown in Figure 1.

A screenshot of a computer

Description automatically generated

Figure 1: Anaconda Prompt to start Jupyter Notebooks

Once Jupyter Notebooks launch, you will see a new web browser page, as shown in Figure 2.

A screenshot of a computer

Description automatically generated

Figure 2: Initial Anaconda Window.

You can now create notebooks, by clicking the **New** button. Note that notebooks are created with a placeholder name and need to be renamed. When creating, you will have a choice of selecting a language. For this course, Python is preferred, but other languages (such as C#) are acceptable.

# Adding Code to Git Hub

One of the outcomes of this exercise is to use GitHub to store versions of the code being used. In this case, our code is simple, as shown in *Figure 3*.

A screenshot of a computer

Description automatically generated

Figure 3: Hello World code

In our simple example we use Flask’s render\_template method to produce the output. This could be replaced with a revision that return the HTML for the homepage directly, however, managing large string of HTML in code becomes cumbersome quickly. The second cell imports the required methods and sets up a rout for the root. The hello\_world method returns the results of render\_template. In this case, the template is simple, as shown in *Figure 4*.

A screenshot of a computer

Description automatically generated

Figure 4: Hello World Index Template

The last piece of this example is to determine if the code is running in Jupyter, or Google Colab. We can determine this by getting the name of the shell, as shown in Figure 5.

A screenshot of a computer program

Description automatically generated

Figure 5: Test if Jupyter

Ensure that you have added your source code to your repository and shared it with your instructor. Also, remember to create an archive (ZIP) of your code and include it in the submission, along with the report.

# Jupyter Lab

An alternative to Jupyter Notebooks is to use Jupyter Labs. Jupyter Notebooks has a simple interface. Jupyter Labs offers additional functionality, beyond notebook editing. In particular, there is a git extension that simplifies version control of notebooks and related files. There is considerable documentation on the topic, including an introductory article (Garcia, 2021). For this course, it is recommended that you use Jupyter Lab, mainly because of the git extension that simplifies code management.

# Conclusion

This document discussed alternatives to creating a development environment for this course. The high-level requirement is that you need an environment where you can specify the version of Python and install required libraries.

# References

Anaconda. (2018). *Getting started with Anaconda Distribution*. <https://docs.anaconda.com/free/anaconda/getting-started/index.html>

Garcia, A. (2021). *Introduction to Using Git in Jupyter*. <https://andgarc.github.io/basic-git-jupyter-lesson/index.html>