Exposition (Beginning - Setting the Stage)

Script: "Hey everyone, I’m finally back after a 6-month break, and I’m super excited to get back to it! Over the last few months, I’ve been balancing a lot—other commitments and my ever-growing interest in Python. If you’re new here, I do Python tutorials, from basic to advanced, along with tips and tricks, and pretty much anything you can think of when it comes to Python programming.

Today, I’m coming back with a bang! In this video, I’m going to showcase all the Python projects I’ve worked on throughout 2024. Setting programming projects for yourself—whether big or small—is such a powerful way to level up your skills, apply theory, and get real-world practice.

Now let’s dive into each project I’ve done this year and the cool things I learned along the way!"

Rising Action (Building the Story - Project Breakdown)

Transition: "I’ve done quite a few projects over the past few months, so I’m going to break them down into categories to make it easier to follow."

Category 1: Maths-Related Projects

"Let’s start with some math-related projects that helped me with complex problem-solving."

Numbers list controller (February)

Script: “In February of this year I developed a small practical tool that can manipulate lists of numbers in multiple ways and store the results in a file when the program is closed automatically. I first made a version as a default command line app where the user chose from an options menu and entered in the numbers how they wanted to manipulate the list. I then also proceeded to make a Tkinter version with a GUI for ease of use. This ‘project’ was just a small fun little maths related tool that I came up with and has been useful since. There weren’t really many challenges in implementing this as it relied on basic file handling and list manipulation techniques that I already knew in Python.”

Goals Challenge Outcome Calculator (September)

Script: “Fairly recently, I had an idea to come up with a calculator for the Footballers Goals challenge that I’ve been watching a lot of recently on YouTube. I regularly watch Razza Mataza failing the 10,000 goal challenge but I also watch other ones and so I thought I’d make a calculator to simulate the best possible, worst possible, average outcomes and more. Also since I had seen others doing a similar thing (albeit without python) and even someone doing it with a custom bot they built, I knew I had to give it a try. My calculator initially included 6 different options for the amount of goals targeted by the challenge, each with their own set of multipliers that the Footballers could be placed in but it quickly expanded as I discovered more goal targets on YouTube. It then sorted the list by ascending order to total the best possible outcome and by descending order to total the worst possible outcome and then outputted them in a user friendly format. It also used the previous history (stored in a text file) to output the amount of attempts so far, the ranking of this attempt, the difference between this score and the average and also current accuracy and average accuracy all in user friendly messages as well. I also introduced an automatic clipboard feature to copy and paste the summary more easily myself then when using the calculator. Overall, it has been quite a profitable project receiving lots of likes on the comments with the summary in them and even directing people from those comments to my own videos on my Fifa Utuber channel. It also helped that the person who coded the custom bot went on holiday and so then I could takeover with the summary statistics then.”

Stay tuned because more exciting projects are coming up…

Complex Random draw (September)

Script: “Another project that I’ve worked on recently that’s maths related is a complex random draw resembling but not limited to the new draw adopted by the UEFA Champions League (and other European competitions) in football. This year the format of these European competitions was changed and it was announced that the draw would be way more complex with most of the work being done by a computer algorithm. This prompted me to make my own version of the draw and challenge myself to do this without any help from web resources, AI generated code or Google searches. So far it’s proving quite the challenge as I’m having to rely solely on my prior programming knowledge and coding ability and it’s still not finished yet, albeit I have done some other things in the meantime and not been concentrating on it that much. I originally planned to finish it before the UCL draw took place at the very start of September but was unable to do so and have been putting it off for the most part since. I do plan to finish it though in the near future.”

Work hours calculator (August)

Script: “Recently I underwent a summer internship that I found through university. During that internship I wanted a way to quickly log my working hours and find out how long I had worked across a number of days that weren’t confined to different days of the week but rather any days I worked like Day 1, day 2, day 10 etc. I ended up making 2 different versions to achieve this with similar features in both. I made a version in Python for my own personal usage and then I also made a version in Microsoft Power apps that I could share with work colleagues.

Category 2: Useful Tools

"Next we’ll move onto some useful tools that can help me and others with tasks, mostly automation tasks "

Random Currency generator (September)

Script: “Recently my Dad tasked with me making a profit/loss calculator for himself. So I made this spreadsheet in Excel. He specifically requested that he wanted a spreadsheet for ease of use. I wanted a quick and easy way of generating the random numbers to fill in currency examples for testing purposes. So I came up with this random currency generator where the user enters the upper and lower bounds for the random generation and how many numbers they want. It then starts to generate the numbers between those specified ranges as decimal or float numbers to 2 decimal places. I used an ingenious method of combining a decimal random generator with the integer one to get decimal numbers in between the range specified. It then also copies the numbers to the clipboard with a 2 second gap to paste the number in between each number. This was very useful for me when filling out the spreadsheet with examples and I can see many other uses for this as well.”

Just a reminder that you can view all these projects on GitHub, especially if you want to use them yourself, either as a tool or study the python code, link is in the description…

Python File Counter (April)

Script: “In April of this year I developed a small tool just in the default python command line that can search a directory and all of it’s sub directories for a specific type of file. This can be improved of course, I can think of 2 or 3 different ways to improve it, such as letting the user enter in the directory path and also maybe having a dictionary mapping of file types to extensions as at the moment the user has to enter the extension of the file but maybe the file type name could be used instead e.g. ‘Microsoft Word’ instead of ‘docx’ but other than that it’s quite a unique idea, I’ve never seen a file counter that counts the amount of files of a specific type in a directory and all its sub directories.”

Leetcode solutions searcher (April)

Script: “Also in April I was completing quite a few Leetcode solutions (probably do a video on Leetcode soon) and I found that it would be useful to have a tool that could search through previous solutions for problems I had completed and then give me the code for that problem and I could also see where I could reuse some code as well. So then I made this where you can just enter in the number of the problem (all problems show a number) such as 2444 and if it exists in my solution bank, outputs the full python code I used for that solution. Pretty handy trick and u guys can try it too, at the moment I don’t have many solutions stored in there but I am working on adding more soon. I could also maybe make a version for other programming languages like Java and JavaScript as well, that could be one for the future, we’ll see”

WhatsApp Poll Configurer (September)

Script: “Fairly recently, I had the issue of having to duplicate WhatsApp polls. So basically there was something that me and my friends were doing in our group chat that required the same or a very similar poll to be sent every week. So I came up with an idea that an image of the poll could be used to either send a new one automatically or a help automate the creation of a poll by periodically providing the user with the copy & paste options from the original poll to speed things up when creating the poll then. It turned out that the first version wasn’t possible as you can’t send a message as a poll to WhatsApp directly through Python. So then I asked ChatGPT for a quick overview and then started coding it. It uses PIL Image libraries and pyTessaract to read the text from the image. It then utilises pyperclip’s copy functionalities to add the options to the clipboard. It also utilises PyAutoGUI’s hotkey functionality to automatically paste it in at intervals. I managed to utilised it then to duplicate a poll perfectly in WhatsApp after some work and testing different methods. However, I tried it with another slightly different poll and this time it didn’t work properly so it probably still needs a bit of work but it’ll get there soon, it’ll be a great general purpose tool then for copying all kinds of polls just by inputting a screenshot via the file path.”

Category 3: Computer science theory:

"Next it’s the penultimate category which involved a couple of projects that helped apply the theoretical side of computer science and make good practical use of it."

Truth Table Generator (March)

Script: “One thing always useful to have when you’re working with logical circuits is a Truth Table Generator. I set about making my own highly advanced version that was user friendly that only incorporated one gate but as many variables as you wanted. The one gate could also be chosen by the user from the 6 standard options: "NOT", "AND", "OR", "XOR", "NOR", "NAND". Then the user could keep entering variables and it produced the table for all possible combinations then with R being the Result or Output column. I also proposed a more advanced version that could handle multiple gates or even a full circuit but that never materialised unfortunately.”

Climax (Peak Moment - Most Exciting Project)

State Machine [Mealy Machine] generator (March)

Script: “Now we come to one of my most exciting and complex projects. I started this in February and it was well into March before I had completed it. You might be thinking it doesn’t sound that complex or complicated but wait till you see this. By generator I mean from scratch kinda thing and I did spend countless hours on this one scratching my head as well. What I wanted to achieve when I first made this was a program that can take the accepted string of a Mealy machine or state Machine with output, and then also an input string and provide the correct output for the string while simultaneously creating the Mealy/State Machine and feeding the input string through it. In university we were learning about State Machines with output, a topic I already had knowledge of and thought it would be a good idea if I could challenge myself to make a python program that generates one from scratch to give the output string for a specific input string. I had never seen anything like this done before, and for good reason, it was insanely difficult to automate something like this without giving more information. In theory, the only information that was required to construct a State Machine was the accepted string, however in practice it was much harder to try and generate one from scratch with only this information. I spent countless hours drawing up diagrams on draw.io and trying different ways to generate the Mealy machine. I had the main logical part down almost instantly, however it was the finer details that stumped me then. The days and weeks went on and although it works, it doesn't always give the correct output string. For instance, if I try and accepted string with 5 digits, the end of the output is slightly wrong. At a time when I had loads of other priorities, I couldn’t continue this then and I never ended up coming back to it. Let me know in the comments if there's anything I can do differently, I've probably already tried it but it's worth a go, who knows you might have a different idea that I never even thought about.”

Category 4: AI and Machine Learning

"Next some projects that are even more exciting than the previous ones, delving now into Artificial Intelligence and Machine Learning, 2 very exciting and now quite popular fields of Computer Science"

Funzy ML (June)

Script: “In June of this year I had a job interview with Funzy and they requested that I do a small example of machine learning in Python as a take home project. I didn’t end up spending too long on it then and 3 or 4 days were given for it. The project was about predicting how many drinks would need to be ordered on a future date based on past sales data. It was just kept simple, there were 4 different drinks that customers could buy and I just made up my own random data and stored them in csv files that could then be processed for the machine learning. It was a step in the direction of AI automation and one that definitely helped my journey in the world of computer science”

AI Cover Letter Writer (September)

Script: “Fairly recently I decided to embark on a task to create my own functional AI application that would also complete a task that would help solve a problem I currently had. So a couple of months ago I gained access to Google’s AI model Gemini in my native python environment and set it up then. I completed a few tests on it beforehand before starting this project then. My access to Gemini was through my Google Cloud workspace subscription that I bought in June on a pay-as-you-go plan. As a developer, I would definitely recommend one of Google Cloud workspace or Microsoft Azure, especially for developing AI applications. I’ve found Google Cloud workspace to be easier to use personally having done a free trial on both, but everyone has their own preference of course.”

AI Testing Agent (August)

Script: “Now coming on to the last project, this is the most complex project and probably took the longest time to complete. I did mention I did a summer internship this summer so during that internship I made this AI testing agent to help and testing some of the software I made so I made it in Python completely and I didn't have much access to Python at the workplace but I could do everything, almost everything in the Google Colab notebook then so I originally made it by using hugging face models publicly available via the transformers library in Python. I didn't really know about this library until I started searching on how I could use AI libraries then in Python so I started off with these transformer library models. I try a few different ones and at one point they were working but in the beginning I got quite a few errors because the input text or the text that the AI would generate output was too large for those models to handle and so I had to change the models multiple times then and it was working eventually but wasn't as accurate I wanted to be. I therefore ended up settling on using Gemini because I mentioned I've been using in my native Python environment recently and so I used the same Google cloud credentials to use Gemini in this project as well. This essentially was an idea I had from like Chat GPT agent builder and there was a version in vertex AI in Google cloud as well you can build an AI agent, it’s like multiple instances of kind of like ChatGPT that all have specific functions then you can call specific functions and they can communicate with each other so I wanted to try and make one in vertex AI but I was struggling so I just decided to go back to python. For the main features, first you input your project name and that can be used for extra context then with the generation then there’s a While loop that asks for the testing question, so like I made a while loop because normally you'll be testing multiple questions for your software at the same time you know so you can test multiple questions so every time you input a question then. When I was testing it then, the questions I was doing was from, so I made an AI search in vertex AI two or three of them AI searches for the workplace then this testing was done using this AI testing agent so I input the question that I put into the search for my testing question then add an uploaded PDF then so these are my document searches so the PDF that the information for the search came from like expected result, so you could upload the pdf document then also upload the screenshot of the actual result so then I run the test of course and the AI starts generating the summary/evaluation. So there’s 2 parts to this, 2 different AI generations then. There’s the first one that just compares the contents of the screenshot and contents of the pdf document in the context of the test and uses the testing question. Then this summary gets passed into the evaluation which is then the second AI generation that looks to evaluate how effective the test was then. If you’ve done software testing, you know how hard it is to think of a summary for your testing report at times, especially if the software your testing is a complex AI search, so this was designed to automate the generation of this summary for a testing report then. It also has copy, paste functionality then to allow you to easily copy and paste the evaluation into your testing report,”

Falling Action (Winding Down - Reflect on Outcomes)

Script: "So after all these projects, I’ve learned a ton. From improving my problem-solving skills with math-related scripts to exploring new AI techniques, 2024 has been a big year for growth as a Python developer. Each project brought its own challenges, but those challenges helped me sharpen my skills."

Outro (Resolution - Looking Forward)

Script: "Wrapping up, I’m really proud of what I’ve accomplished this year. But this is just the beginning. I’m planning to develop more AI applications in the future, starting with expanding on the AI Testing Agent and the AI Cover Letter Writer.

In my upcoming videos, I’ll be working on a fun Leetcode challenge and even doing a ChatGPT vs Copilot comparison using Gemini—so be sure to like and subscribe if you don’t want to miss that! Also, let me know in the comments which project was your favourite or what projects you’re working on right now. And of course, feel free to give me ideas for new projects to try out."

"Thanks for watching, and I’ll see you in the next one!"