**Pytomation: Learn from the Basics to Work Automation**

**Introduction**

**Project Context**

The issue with self-learning is it is inefficient. Most of the problem students face when they decide to self-learn is their lack of knowledge on what they should learn their lack of time and lastly their motivation and interest to learn.

The reason the student is lacking the knowledge on what they should learn is simply they do not know where to start, or where to progress their learning. There are many resources on the internet that the student can start learning, and making decisions in learning any of them is quite difficult, especially since there are many resources that are either irrelevant or not the appropriate skill level for the student’s progress. It is the lack of direction in teachers or tutors provides that the student lacks when it comes to self-learning which makes self-learning inefficient if the student does not know how to start or progress.

In the following issue, the reason why the student lacks time correlates with the first issue stated. The student learns too many educational materials that are either irrelevant or not part of the topics the student should be learning, therefore, many students waste time and effort. Another reason is if the students have other more important tasks, what if the students have schools to attend that do not relate to the topics in these students’ self-teaching? What if the students have work, and cannot afford to waste a lot of time?

While students can learn at their pace, and without any competition and deadline to comply with, these students tend to lose focus on their learning, and mostly lose their motivation to learn after losing that focus, especially if they student has more important things to deal with.

With the current technology, people are not aware of how it can change the way that they live especially how they work. As technology immediately evolves and improves, it cannot keep up with its latest trends. People who use technology like computers especially the student for their schoolwork, it is exhausting to do a lot of tasks with the same actions repeatedly, many people who study technology does not aware that it can do more things like automating different tasks that repeatedly doing which makes the work done in a short time. Most people will not know or believe something until someone told them and prove to them that it is possible to happen.

This study aims to help students, particularly those with a lot of work and activities, learn a programming language that can enable them to automate their daily tasks. By doing so, they can free up more time to focus on personal development after completing their required tasks. Most of the students does not aware of the potential of different languages because maybe they are lazy to explore and does not aware of how many things these programming languages can do for their lives. Learning how to automate things can save up so much time in their daily life.

**Objectives of the Study:**

**Create a Student Data Recorder, and Progression system**

* Create a Skill Assessment System
* Create a Quiz Maker/Giver System
* Create a Lecture Giver System
* Create a Code Checker
* Teach people especially students how to automate things using python

**Create a Student Data Recorder and Progression System:**

* Create a database that stores student data for skill scores
* Create Progression Levels for classifying lectures and quizzes, and the student’s skills.
* Create a score calculator that assesses the skill level of the student. It should set the skill level of the student base on their scores on each skill criterion and limit their topic options to only the appropriate topics for their level.

**Create a Skill Scorer System:**

* Create a scoring system that scores code errors and code test retries.

**Create a Quiz Maker System:**

* Create a quiz that contains variables where it can be randomly generated in creating similar, but unique quizzes.
* Create unique tasks for each topic that makes the student use the skills acquired from the lectures.

**Create a Lecture Giver System:**

* Create interactable example demos

**Create a Code Checker:**

* Create a string reader that reads the code and compares statements for creating corrections if mistakes were made.

**Scope and Limitations of the Study**

This project will be teaching python console programming, to people who are interested in programming, within the amateur and beginner skill levels. The program will provide lectures and quizzes for proper console coding in python language and teach the student the industry standard in coding with python. The program will also be providing activities, which the student may find useful, such as tasking the student to create their scripts for automating tasks. The main goal of this study is to teach people how they can use the Python programming language to automate their tasks and make their life easier.

Features:

**Core Function Features:**

**Learning Progression:**

* It will assess the skill of the user.
* It will suggest quizzes and lectures based on the user’s skill level and progress.
* Each learning criterion will be separated by modules and the user must finish the prerequisites or the basic modules to progress.
* It will allow the user to choose modules that are on their level, but lock modules that are too high for them.

**Randomized Quiz:**

* It will create a quiz with randomized problems and randomized correct answers.
* It will create different quizzes and activities when the user retries the quiz.

**Interactable Lecture Demonstrations:**

* Every module will provide the user detailed description of the topic and give instructions on how to use them.
* It will provide interactable demonstrations of codes and gives more detailed information about the code parts when the user hovers their mouse over it.

**Practice mode:**

* It allows the user to use the program as Python IDE and allows them to Test and Practice codes that they have learned to do.
* It compiles the user’s code and runs it.

**Miscellaneous Activities:**

* There are random activities (for fun mostly) that will randomly and rarely replace a quiz whenever a quiz is occurring. It will be a simple game or puzzle that focuses on the current module.
* If the user finished the module, they will then be able to play all the activities on that specific module.

**Personalization:**

* The GUI objects will be movable, and the user can place each object in areas of preference.
* There will be a light and dark theme.

**Search Bar:**

* a search bar that will find things that the user typed.
* it will also suggest similar and correlated topics to what the user typed.

**Code/Application Examples and Challenges:**

* after completing the Course, the user will unlock code and application examples which will display source codes of simple applications such as a standard calculator and simple word juggler.
* the user will also have the option to do challenges which will task the user to create console applications themselves, and task them to automate things or make simple machine learning algorithms.

**Limitations:**

This program will not be able to teach other programming languages other than python console, and the learning resources are only about python console. The program will only be usable in a Windows PC environment. This will only teach the basics of the syntax of python coding there used industry-standard programming but have only a limited understanding of certain concepts, it will not teach high-level techniques and industry-standard design patterns. It cannot assess the true skill of a user because the system will base only on the given criteria and the difficulty varies from the experience of the user.