DOCUMENTATION

INTRODUCTION

This is Oxford Quantum circuit -> OQC's string ingestion API documentation done by developed by BIOKU. This documentation provides a brief overview of the OQC's string ingestion API, what it is and what it does. The overview of this API is to take input string from the user, check if the string is pattern is a valid entry, if it is a valid entry, the system therefore processes it and return feedback to the user. You can read more

You will be able to get started with cloning this project, get the required dependencies to make the project work on your own local machine and also see a detailed walkthrough of what each unit of the codebase of this project is doing

Getting started with this project requires a couple of easy to follow steps we need to set up our environment –setting up environment simply means to create a unique place for the project.

It is a good idea to create a virtual environment for each project. Although, this is not a compulsory requirement, but doing so helps to have a project with its unique dependencies and packages

–so, Let us create a virtual environment step

step 1 create a folder <name> – in this case, example folder name to create is OQC_TEST open the command line and navigate into this folder.

step 2 install the virtual environment package by running this command $ pip install virtualenv

step 3 create virtual environment by running this command $ virtualenv env

NOTE -> Kindly note that the env is the name of the virtual environment that you want to create. It is recommended to use a simple name. it is a common practice to use env which means environment

Step 4 activating the virtual environment that we have just created run the command $ env\Scripts\activate if this command executes successfully, the path in your command line will be prefixed with (env)

We have now successfully created our virtual environment

FLASK INSTALLATION

Flask is a light weight python framework for web development. To get started with flask, it needs to be installed

Step 1 execute the command $ pip install flask

step 2 Confirm that flask is successfully installed by running the command $ flask –version

List of Dependencies

aioflask 0.4.0

Flask-WTF 1.0.1

SQLAlchemy 1.4.37

This project's functionality is dependent on the list of these packages

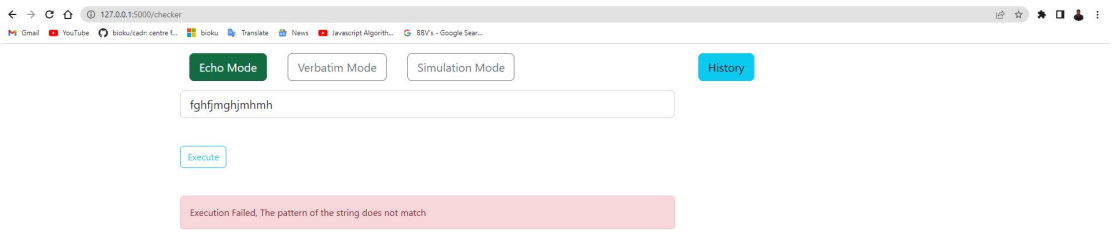
1. aioflask -> This package makes it possible to write asynchronous program in python to install aioflask execute the command $ pip install aioflask

2. SQLAlchemy -> SQLAlchemy is a database package that makes it possible to store information in flask to install SQLAlchemy, execute the command $ pip install SQLAlchemy

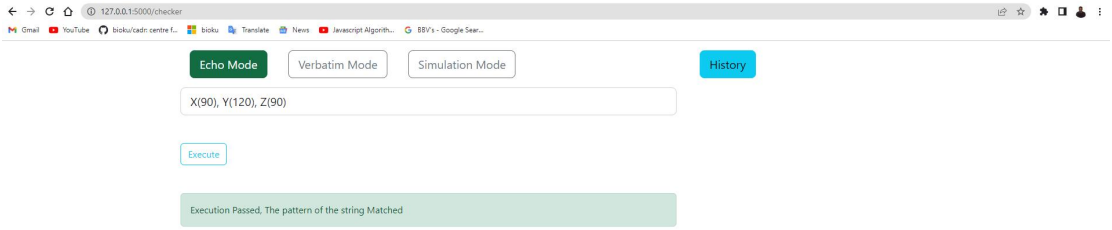
3. flask-wtf -> flask-wtf is a package that makes it easy to build form easily in flask to install flask-wtf, execute the command $ pip install flask-wtf

Functionalities:

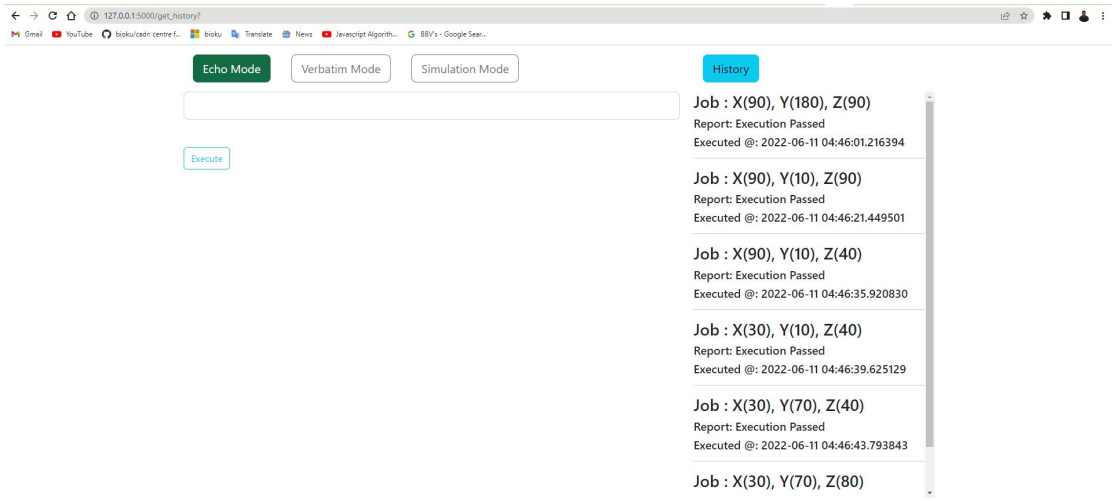
User input: User Inputs a pattern of string in the text field
Checker Function : The checker function takes the user input, reads it and check if it matches the pattern of the system, If it matches, the system instantiates a runtime class and passes the string to the execute function, else, the system returns error feedback to the user.



Execute function: The executes function takes the job, and store it in the database and return a successful execution message to the
user.



User Interface: The user Interface of the system has 3 primary mode buttons, 1 button to query the database for the history of successful jobs. 1 action button that posts the job to the app's engine for processing and the user input text field that allows the user to field in a job



LIST OF FILES

4. index.html

5. index.css

6. __init__.py

7. routes.py

8. runtime.py

9. db.py

10. input_form.py

11. models.db

12. test_execute.py

13. run.py