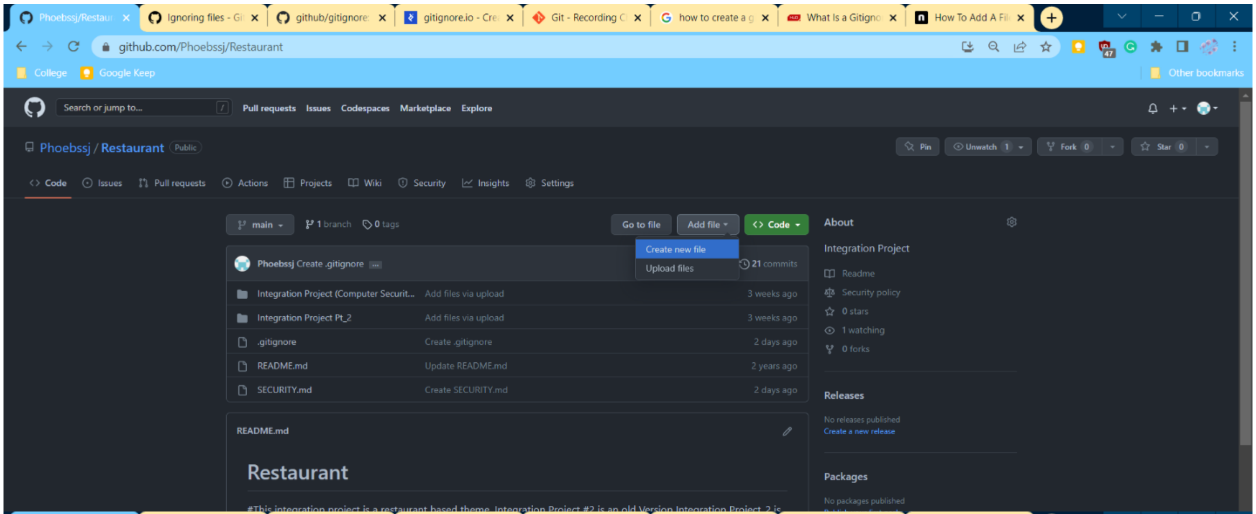
Project Check in Progress:

Focused on 2) To protecting sensitive files by marking them as private through GitHub.

Progress:

- Currently added a .gitignore file to the main branch of the repository to hide all text files used for the project. The research for finding out how to do the .gitignore within the GitHub repository was difficult to find, as my project was never linked to GitHub. From the prior check in, one of the many focused way to create and add a .gitignore was through a terminal My project was create on a prior desktop and was never committed to github through the IDE nor do I have git installed either, therefore the regular tutorial that github provides wouldn't work. Another source provides a bit of insight on how to create it, and it was done by creating a new file on github and name it .gitignore and provides the template for the file type it's being referenced to and include the files within the .gitignore and commit the changes.



github.com/Phoebssj/Restaurant

Phoebssj / Restaurant Public

main · 1 branch · 0 tags

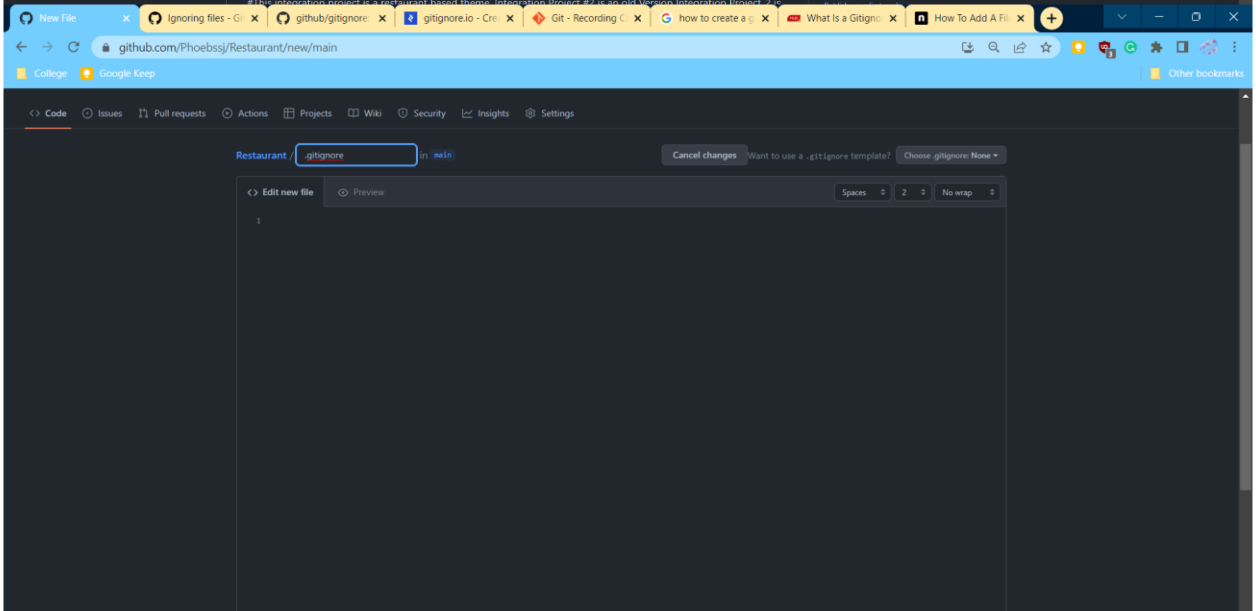
Phoebssj Create .gitignore · 21 commits

Integration Project (Computer Securit...	Add files via upload	3 weeks ago
Integration Project Pt_2	Add files via upload	3 weeks ago
.gitignore	Create .gitignore	2 days ago
README.md	Update README.md	2 years ago
SECURITY.md	Create SECURITY.md	2 days ago

README.md

Restaurant

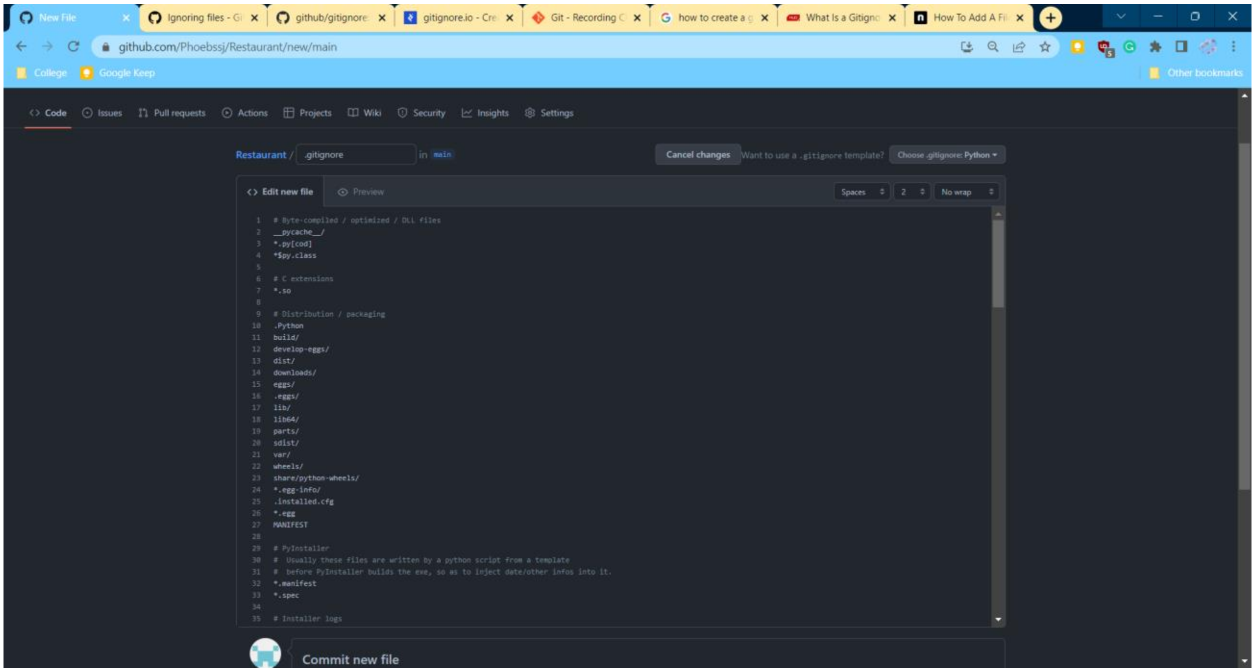
#This integration project is a restaurant based theme. Integration Project #2 is an old Version Integration Project 2 is



github.com/Phoebssj/Restaurant/new/main

Restaurant / .gitignore in main

Want to use a .gitignore template? Choose .gitignore: None

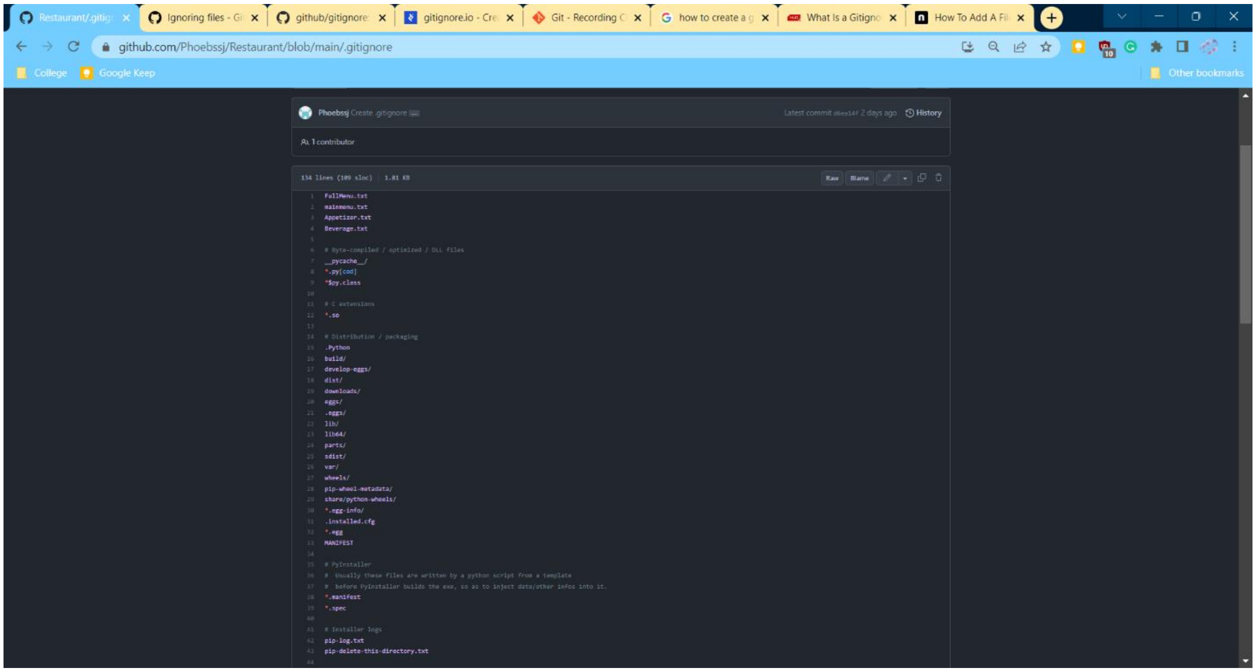


Restaurant / .gitignore in main

Want to use a .gitignore template? Choose .gitignore: Python

Spaces 2 No wrap

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST

# PyInstaller
#  Usually these files are written by a python script from a template
#  before PyInstaller builds the exe, so as to inject date/other infos into it.
*.manifest
*.spec

# Installer logs
```



Phoebssj Create .gitignore

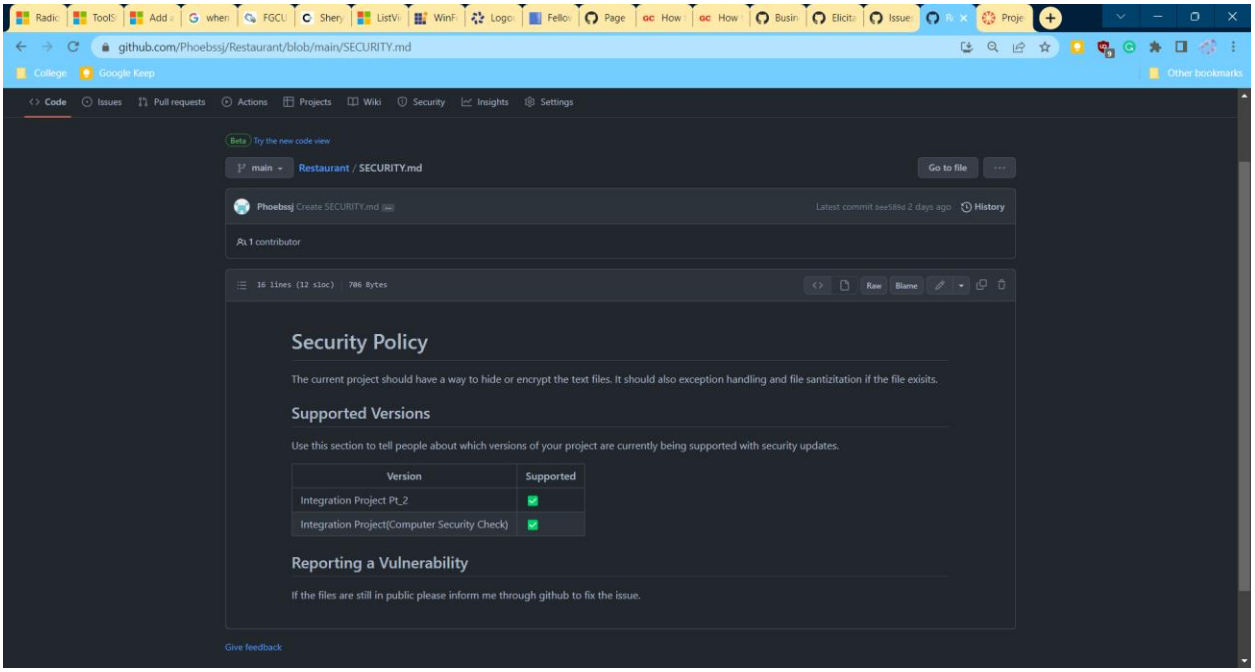
Latest commit d6ea14f 2 days ago

1 contributor

134 lines (109 sloc) 1.81 KB

```
FullMenu.txt
mainmenu.txt
Appetizer.txt
Beverage.txt

# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
pip-wheel-metadata/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST

# PyInstaller
#  Usually these files are written by a python script from a template
#  before PyInstaller builds the exe, so as to inject data/other infos into it.
*.manifest
*.spec

# Installer logs
pip-log.txt
pip-delete-this-directory.txt
```

- Added a Security MD for the explanation referring to the integration project, in which there was not much to add as the specific condition for it to be tampered with is if I provide access to the repository, and for authentication purposes, my account has Two-Factor Authentication. Hence the reason for the policy being very simplistic. I plan to add more on the policy for any further testing that can be found in the project that it could be tampered with.



Security Policy

The current project should have a way to hide or encrypt the text files. It should also exception handling and file santizitation if the file exisits.

Supported Versions

Use this section to tell people about which versions of your project are currently being supported with security updates.

Version	Supported
Integration Project Pt_2	✅
Integration Project(Computer Security Check)	✅

Reporting a Vulnerability

If the files are still in public please inform me through github to fix the issue.

Focused on 8) Accounting: (exception handling, errors, and user input fails)

Progress:

- Still currently working on try except of negative integers getting through. From the prior check they are robust in handling exceptions for both negative integers as well as other data types. Though a few negative integers do get by. So, I'm planning to rewrite the checks to properly take in negatives properly.

When using an Online repository like GitHub, it restricts the amount of access that users have to a repository. GitHub has the ability to hide a repository that shouldn't be shown to the public and a solution can would be creating a .gitignore file to specifically hide certain files included within the repository. In addition, GitHub tracks any commits made to the repository, and any case of traffic made on the repository.

Testing:

Currently no test has been implemented but will be implemented in the future. I intend to do testing once I re-write the error made within the code. It would go through a case of unit testing to see if error handling is still a current issue. In addition will also test for accounting with logging read files and user input handling.

Issues I will try to attempt:

1) Attempt to fix the handling errors where they end up accepting negative integers.
2) Encrypting the text files

Currently still working on:

1) Re-writing the code in such a way that it takes the user inputs f=though functions and verifies there input to be current.
2) To sanitize the text files within the code.

Sources Used:

https://www.stackhawk.com/blog/command-injection-python/

https://rules.sonarsource.com/python/RSPEC-2737

https://resources.infosecinstitute.com/topic/python-language-basics-understanding-exception-handling/

https://snyk.io/blog/the-dangers-of-assert-in-python/

https://docs.python.org/3/tutorial/errors.html#predefined-clean-up-actions

https://docs.python.org/3/tutorial/errors.html

https://www.techtarget.com/searchsecurity/feature/Exception-handling-best-practices-call-for-secure-code-design

https://docs.python.org/3.5/library/itertools.html

https://blog.sqreen.com/developer-security-best-practices-how-to-validate-input-data-in-python/

https://problemsolvingwithpython.com/08-If-Else-Try-Except/08.05-Try-Except-Statements/

https://www.dremendo.com/python-programming-tutorial/python-if-elif-questions/q2-check-number-is-positive-negative-or-zero-in-python

https://pynative.com/python-check-user-input-is-number-or-string/

https://programming-21.mooc.fi/part-6/3-errors

https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files#excluding-local-files-without-creating-a-gitignore-file

https://github.com/github/gitignore

https://nira.com/how-to-add-a-file-to-gitignore-on-github/

https://stackoverflow.com/questions/63972265/create-a-gitignore-file-after-creating-the-repository-and-publish-idea-file

Sources recommended by Professor:

https://www.pythonforbeginners.com/control-flow-2/when-to-use-try-catch-instead-of-if-else

https://docs.python.org/3/library/exceptions.html

https://docs.python.org/3/library/stdtypes.html#str.isdigit