**Vulnerability Rules Portal Documentation:**

open the command prompt and change the current path to project folder

Then run the command node server.js

**Installation part:**

npm install --save

Then, Change the postgres database credentials in server.js

**create database and tables:**

create database name;

1)CREATE TABLE cve

(

name text NOT NULL,

status text,

description text,

reference text,

phase text,

votes text,

comments text,

created\_at text,

cwe text,

technology text,

frame\_work text,

nist text,

cwer text,

sample\_code text,

remediation text,

cast\_support text,

cast\_pm\_comments text,

rd\_comments text,

consultant\_comments text,

rem\_links text,

CONSTRAINT cve1\_pkey PRIMARY KEY (name)

) ;

2)CREATE TABLE castquality

(

id text,

originalname text,

technology text,

tag text,

standard text

) ;

3)CREATE TABLE checkmark

(

language text,

packagename text,

queryname text,

cweid text

) ;

4)CREATE TABLE cisq

(

cisqid text NOT NULL,

descriptor text,

remediation text,

CONSTRAINT cisq\_pkey PRIMARY KEY (cisqid)

) ;

5)CREATE TABLE cwe

(

cwe\_id text,

name text,

weakness\_abstraction text,

status text,

description text,

extended\_description text,

related\_weaknesses text,

weakness\_ordinalities text,

applicable\_platforms text,

background\_details text,

alternate\_terms text,

modes\_of\_introduction text,

exploitation\_factors text,

likelihood\_of\_exploit text,

common\_consequences text,

detection\_methods text,

potential\_mitigations text,

observed\_examples text,

functional\_areas text,

affected\_resources text,

taxonomy\_mappings text,

related\_attack\_patterns text,

notes text,

created\_at text,

cwe\_sans\_top\_25\_2009 text,

cwe\_sans\_top\_25\_2010 text,

owasp\_top\_10\_2004 text,

owasp\_top\_10\_2007 text,

owasp\_top\_10\_2010 text,

owasp\_top\_ten\_2013 text,

sei\_cert\_cpp\_coding\_standard\_2016 text,

the\_cert\_c\_secure\_coding\_standard\_2008 text,

the\_cert\_oracle\_secure\_coding\_standard\_for\_java\_2011 text,

created\_at text,

examples text,

technology text,

frame\_work text,

nist text,

cve text,

sample\_code text,

remediation text,

cast\_support text,

cast\_pm\_comments text,

rd\_comments text,

consultant\_comments text,

rem\_links text

) ;

6)CREATE TABLE fortify

(

title text,

abstract text,

explanation text,

example text,

reference text,

technology text,

owasp text,

stig text,

nist text,

cwe text,

pci text,

technical\_guidelines text,

disa text,

fips text,

gdpr text,

wasc text,

cwe\_id text

) ;

7)CREATE TABLE nist

(

domain text,

"number" text,

title text,

priority text,

impact text,

instructions text,

guidance text,

enhancements text

) ;

8)CREATE TABLE pci

(

did text,

domain text,

rid text,

requirement text,

cid text NOT NULL,

control text,

test text,

guidance text,

pri\_6 text,

pri\_3 text,

castapplicable text,

CONSTRAINT pci\_pkey PRIMARY KEY (cid)

) ;

9)CREATE TABLE stig

(

vuln\_id text,

rule\_id text,

group1 text,

version text,

severity text,

title text,

description text,

diacap text,

rmf text,

cci text,

check\_id text,

check\_text text,

fix\_id text,

fix\_text text,

filename text,

technology text,

frame\_work text,

nist text,

cwe text,

sample\_code text,

remediation text,

cast\_support text,

cast\_pm\_comments text,

rd\_comments text,

consultant\_comments text,

rem\_links text

) ;

**Go to portal:**

http://skecastvm:2281/ (For VM)

localhost:4900 (For Local Machine)

**To Upload the data:**

Sign in to portal with username:admin and password:cast and then Start Uploading the csv files in portal

Note:To add data to corresponding table, csv file should be same as table name. Also header names in csv and in table should be same.

After uploading all standards data, the portal is ready to use.

**Technologies Used to build the portal:**

Node.js (For portal)

Python (for scraping)

Postgresql (For database storage)

HTML

EJS

**GitHub Code Structure:**

Standards repository

src

public

css

Istyle.css

style.css

sview.css

img

index.html

public.html

search.html

search1.html

index.html

style.css

uploads

.csv files

views

.ejs files

.editorconfig

.gitignore

README.md

angular.json

package-lock.json

package.json

server.js

tsconfig.json

tslint.json

**Steps to scrape:**

Step 1: Go to Fortify Website:

https://vulncat.fortify.com/en/weakness

Step 2: Apply Web Scraping on the HTML pages using Python Beautiful Soup Library.

Step 3: Write Code to push the details to the Excel file.

Step 4: Push the data to PostgreSQL Database.

Step 5: Then, Fortify data is ready to be used in the portal.