ContentNotifier

**“ContentNotifier”**

**“An application to send a notification through an Email”**

**A Project Report Submitted In Partial Fulfillment Of The Requirements To Rajiv Gandhi University of Knowledge Technologies Srikakulam**

**For The Award Of Degree In**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**L. Sivasri (s180655)**

**K. Devika (s180632)**

**V. Divya (s170180)**

**Under the Esteemed Guidance of**

**Mr.T. Anil Kumar, Assistant Professor**

****

**Department of Computer Science and Engineering,**

**RGUKT-SRIKAKULAM, ECHERLA.**

**April 2023**

Andhra Pradesh, India

ContentNotifier



**CERTIFICATE**

This is to certify that the thesis work entitled **“ContentNotifier”** was successfully completed by **L.Sivasri (s180655), K.Devika (s180632), V.Divya (s170180)** in partial fulfillment of the requirements for the Mini Project in Computer Science and Engineering of Rajiv Gandhi University of Knowledge Technologies under my guidance and output of the work carried out is satisfactory.

**Mr. T. Anil Kumar, Asst. Prof Mr. N. Sesha Kumar, Asst.Prof Project Internal Guide, Head of the Department, Department of CSE, Department of CSE, RGUKT-SKLM RGUKT-SKLM**

**i**

ContentNotifier



**BONAFIDE CERTIFICATE**

Certified that this project work titled **“ContentNotifier”** is the Bonafide work of **L.Sivasri (s180655), K.Devika (s180632), V.Divya (s170180),** who carried out the work under my supervision, and submitted in partial fulfillment of the requirements for the award of the degree, BACHELOR OF TECHNOLOGY, during the year 2022 – 2023.

**Mr. T. Anil Kumar, Asst. Prof Mr. N. Sesha Kumar, Asst.Prof Project Internal Guide, Head of the Department, Department of CSE, Department of CSE, RGUKT-SKLM RGUKT-SKLM**

**ii**

ContentNotifier

**DECLARATION**

We hereby declare that this thesis entitled “ContentNotifier” an application is carried out by us during the academic year 2022-2023 in partial fulfillment of the requirements for the Mini Project in **Computer Science and Engineering.**

We further declare that this dissertation has not been submitted elsewhere for any Degree. The matter embodied in this dissertation report has not been submitted elsewhere for any other degree. Furthermore, the technical details furnished in various chapters of this thesis are purely relevant to the above project and there is no deviation from the theoretical point of view for design, development and implementation.

Date :

Place :

**Project Associate**

**L. Sivasri (s180655)**

**K. Devika (s180632)**

**V. Divya (s170180)**

**iii**

ContentNotifier

**ACKNOWLEDGEMENT**

We would like to articulate my profound gratitude and indebtedness to my project guide **Mr. T. Anil Kumar sir(Asst. Prof),** Assistant Professor who has always been a constant motivation and guiding factor throughout the project time. It has been a great pleasure for me to get an opportunity to work under his guidance and complete the thesis work successfully.

We wish to extend my sincere thanks to **Mr. N. Sesha Kumar sir(Asst.Prof),** Head of the Computer Science and Engineering Department, for her constant encouragement throughout the project.

We are also grateful to other members of the department; without their support my work would have been carried out so successfully.

We thank one and all who have rendered help to me directly or indirectly in the completion of our thesis work.

**Project Associate**

**L. Sivasri (s180655)**

**K. Devika (s180632)**

**V. Divya (s170180)**

**iv**

ContentNotifier

**ABSTRACT**

Time is a valuable asset to human beings. Managing the time well, will let the people do multiple jobs intime. In today’s digital world, each and every work is virtualized and is being accessed through websites using the internet. The time required to know the presence of required data in a digital asset is quite hard. Manual checking of data might need more time and focus. In order to facilitate people with an easy way for data checking in the digital assets, this **“ContentNotifier”** project aims to provide a platform where the users can get the email notifications for the data they are searching for. This mainly helps the people to know the presence of data without checking manually. The software implemented will check for the data instead of the people who actually want the data. Those who are waiting for the upcoming data in corresponding webpages can use this platform. This project could make people comfortable in searching the data in digital assets.

**v**

ContentNotifier

**INDEX**

**CH. NO CONTENTS PG.NO**

**1 INTRODUCTION 1-2**

1.1 Introduction 1

1.2 Problem of the Statement 1

1.3 Objective 1

1.4 Scope 2

1.5 Applications 2

1.6 Limitations 2

**2 LITERATURE SURVEY 3**

2.1 Collecting Information 3

2.2 Study 3

2.3 Benefits 3

2.4 Summary 3

**3 SYSTEM ANALYSIS 4-5**

3.1 Existing System 4

3.2 Disadvantages 4

3.3 Proposed System 4

3.4 Advantages 4

3.5 System Requirements 5

**4 SOURCE CODE 6-10**

4.1 projectFile.py 6-7

4.2 Interface.py 8-10

**5 SYSTEM TESTING 11-13**

5.1 Introduction 11

5.2 Types Of Testings 11-12

5.3 Levels Of Testings 13

**6 CONCLUSION 14**

**7 FUTURE ENHANCEMENT**  **14**

**8** **APPENDIX 14**

**REFERENCES**

ContentNotifier

**Chapter-1**

**INTRODUCTION**

**1.1 Introduction**

Time is a valuable asset to human beings. Managing the time well, will let the people do multiple jobs intime. In today’s digital world, each and every work is virtualized and is being accessed through websites using the internet. The time required to know the presence of required data in a digital asset is quite hard. Manual checking of data might need more time and focus. In order to facilitate people with an easy way for data checking in the digital assets, this **“ContentNotifier”** project aims to provide a platform where the users can get the email notifications for the data they are searching for. This mainly helps the people to know the presence of data without checking manually. The software implemented will check for the data instead of the people who actually want the data. Those who are waiting for the upcoming data in corresponding webpages can use this platform. This project could make people comfortable in searching the data in digital assets.

**1.2 Statement of the Problem**

Manual data checking in digital assets is time-consuming and inefficient, requiring constant monitoring and focus. There is a need for an automated solution that can proactively search for specific data and notify users when it becomes available, simplifying the process and saving time.

**1.3 Objective**

● To provide a notification of upcoming data through an email..

● To reduce the time required for manual data checking.

1

ContentNotifier

**1.4 Scope**

● **Target users**: This project aims to give content updates via email to a wide range of people who are waiting for upcoming data.

● **Accessibility**: This project aims to give notification of searching data which is more accessible to a wide range of people. This includes providing free access to the platform.

**1.5 Applications**

* To know updates in websites which do not contain the feature of user notifications.

**1.6 Limitations**

This project works well for the website which supports web scraping.

2

ContentNotifier

**Chapter-2**

**LITERATURE SURVEY**

**2.1 Collect Information**

We have taken the information from the other sources like land-based websites (99acres), some of the online hotel websites and other sources to check how they are categorized and organized and we proposed this ourselves. From that we collect the price for the particular service.

**2.2 Study**

**ContentNotifier key features:**

● It takes less time.

● User Friendly .

● Simple and fastly notified when the data is updated.

**2.3 Benefits**

● To provide a notification of upcoming data through an email.

● To reduce the time required for manual data checking.

**2.4 Summary**

“ContentNotifier” will help people with a system which is to provide the platform where the users can get the email notifications for the data they are searching for from a particular website. This mainly helps the people to know the presence of data without checking manually. Those who are waiting for the upcoming data in corresponding webpages can use this platform.

3

ContentNotifier

**Chapter-3**

**System Analysis**

**3.1 Existing System**

# LexisNexis Content Notifier website can be found on the internet.

# 

# **3.2 Disadvantages**

● It can gather information by checking the details on various websites.

● It can be more time-consuming for checking multiple websites.

**3.3 Proposed System**

We proposed a system which is to provide the platform where the users can get the email notifications for the data they are searching for from a particular website. This mainly helps the people to know the presence of data without checking manually. Those who are waiting for the upcoming data in corresponding webpages can use this platform.

**3.4 Advantages**

● **Time Saving:** Faster results will be produced as notifications to your Gmail Account.

● **Easy to get the update**: This will be used to easily get the content from the webpage.

**3.5 Disadvantages**

* Works for some websites only which support scrapping.

4

ContentNotifier

**3.6 System Requirements**

**Software Requirements:**

• PYTHON

• Operating System – Windows(version 10)

• Editor – Jupyter Notebook(Anaconda)

**Hardware Requirements:**

• RAM: 4GB

• Hard disk: 500 GB above

5

ContentNotifier

**Chapter-4**

**Source Code**

**4.1 projectFile.py**

# importing necessary libraries

import urllib.request

import re

import smtplib

from functools import partial

import schedule

file\_path = r"C:\Users\SIVASRI\Desktop\emails.txt"

with open(file\_path, 'r') as file:

f = file.read()

def check(web\_src,required,r\_mail):

page=urllib.request.urlopen(web\_src)

data=page.read().decode("UTF-8")

data=data.lower()

results = re.findall(required.lower(),data)

if (len(results)!=0):

# Sender and recipient email addresses

sender\_email = 's180655@rguktsklm.ac.in'

recipient\_email = r\_mail

# SMTP server settings for Gmail

smtp\_server = 'smtp.gmail.com'

smtp\_port = 587

sender\_password = 'sivasri@1907'

6

ContentNotifier

server = smtplib.SMTP(smtp\_server, smtp\_port)

server.starttls()

server.login(sender\_email, sender\_password)

# Send the email

server.sendmail(sender\_email, recipient\_email, f'The data {required}, you are looking for is just arrived. Check now here: {web\_src}')

print("Emil sent successfully")

with open(file\_path, 'w') as file:

file.write(f'{recipient\_email}-{required}')

# Close the connection to the SMTP server

server.quit()

else:

print('not found')

r\_mail = input("Reciever mail id:\t")

required = input("Required data:\t")

web\_src = input("Web source link:\t")

if(re.fullmatch(re.compile(r'([A-Za-z0-9]+[.-\_])\*[A-Za-z0-9]+@[A-Za-z0-9-]+(\.[A-Z|a-z]{2,})+'),r\_mail)):

try:

page=urllib.request.urlopen(web\_src)

schedule.every(10).seconds.do(partial(check,web\_src,required,r\_mail))

while(f'{r\_mail}-{required}' not in f):

schedule.run\_pending()

with open(file\_path, 'r') as file:

f = file.read()

except:

print("Provided web source isn't accessible.")

7

ContentNotifier

else:

print("Invalid email id.Please enter a valid one.")

**4.2 Interface.py**

import urllib.request

import re

from tkinter import \*

# from projectFile import check

def show():

if(re.fullmatch(re.compile(r'([A-Za-z0-9]+[.-\_])\*[A-Za-z0-9]+@[A-Za-z0-9-]+(\.[A-Z|a-z]{2,})+'),rm.get())):

res = 0

try:

urllib.request.urlopen(ws.get())

except:

res = 1

if(res==1):

op.config(text='Sorry the website data can\' be accessed')

ws.delete(0,END)

rd.delete(0,END)

rm.delete(0,END)

else:

op.config(text='Added Successfully.........')

else:

ws.delete(0,END)

rd.delete(0,END)

rm.delete(0,END)

8

ContentNotifier

op.config(text='Something went wrong.......')

root = Tk()

root.title('Application Page...')

Label(root,text="ContentNotifier",fg="Orange",font=("Helvetia",15,"bold","italic"),height=10).grid(row=-0,column=14)

Label(root,text="\tThis tool will inform you through the email regarding your required data ",font=20).grid(row=1,column=15)

Label(root,text="based on the parameters you provided here.",font=20).grid(row=2,column=15)

Label(root,text="ADD YOURSELF NOW .....",font=20).grid(row=3,column=15)

Label(root,text="WELCOME ",width=30,font=35).grid(row=1,column=0)

Label(root,text="To the content notifier tool.... :",font=35).grid(row=2,column=0)

l=Label(root,text='Web Source : ' , fg='Indigo' ,font=40)

l1=Label(root,text='Required Data : ' , fg='Indigo' ,font=40 )

l2=Label(root,text='Recipient Mail: ' , fg='Indigo' ,font=40 )

ws=Entry(root)

rd=Entry(root)

rm=Entry(root)

l.grid(row=1,column=10)

l1.grid(row=2,column=10)

l2.grid(row=3,column=10)

ws.grid(row=1,column=13)

rd.grid(row=2,column=13)

rm.grid(row=3,column=13)

9

ContentNotifier

op=Label(root)

op.grid(row=6,column=13)

res=Button(root,text="SUBMIT",command=show, fg='Magenta',bg='AliceBlue',width=10,height=2,font=12)

res.grid(row=7,column=13)

root.mainloop()

10

ContentNotifier

**Chapter-5**

**SYSTEM TESTING**

**5.1 Introduction**

Testing is a kind of procedure, which is used to check the cases in which the result is client centric or system centric. There will be some cases where it is completely system centric i.e., the results will not be in the favor of the client, which leads to the Fault Case. Fault cases are the type of scenario where it doesn’t show the result based on the user requirement. Basically, in the testing phase there will be several types of testing i.e., Unit Testing, Integration Testing and Functional Testing. In this project, we have taken some cases to test the output of the program.

**5.2 Types Of Testings**

**Unit testing:**

Unit Testing is a type of testing, in which the testing is done individually(block wise). This will result in calculating the test cases unit wise, which is validation of web source, gmail, sending gmail, scraping the webpage,automating the script in our project. We have tested the smallest unit of our application. This is done by the developer during the developing phase, to check each condition, whether it is working correctly based on the client requirements. If in that scenario, an error occurs, it is easy to identify and correct it then and there itself.

**Integration testing:**

Integration Testing is a type of testing in which it is done as a collection of single unit test cases. We need to integrate the individual test cases as a single block of test cases and then we need to test that entire, so that we are able to get that integrated test case results. In this project basically, we have tried different unit test case samples. By integrating those unit test cases, we can be able to do the integration testing as well.

11

ContentNotifier

**Functional testing:**

Functional Testing basically checks whether the developed application can be able to reach its heights in satisfying the client requirements by delivering the required functionality of the application. This type of testing is done by the Quality Assurance team, in which they do the Black box testing. It is only considered for the Validation purpose.

Invalid enter : recognized lessons of unacceptable effort must be rejected.

Capabilities : recognized features ought to be exercised.

Output : recognized courses of software outputs have got to be exercised. Systems/Procedures : performance of the system here was invoked.

**System testing:**

System Testing ensures that it is totally tested after integration testing level. It is one of the most important parts of the software development life cycle (SDLC), as it concerns providing a rough structure on how the application software is working or not.

**White Box Testing:**

Difficult to ensure that the whole included agenda process meets principles. It examines a pattern to make sure an identified and predictable outcome. An illustration of procedure testing is the configuration oriented approach integration scan. System testing is based on approach descriptions and flows, emphasizing pre-driven system links and integration aspects.

**Black Box Testing:**

This is testing the software with none advantage of the inside workings, establishment or words of the unit life form veteran.

12

ContentNotifier

**5.3 Levels Of Testings**

**Unit testing strategy:**

Unit checking out is most commonly performed as a part of a mixed code and unit experiment part of the software lifecycle, though it is not exceptional for coding and unit checking to be performed as two targeted phases.

Test strategy and approach:

Field testing can be carried out manually.

Test objectives:

Each validation should work correctly.

**Integration testing strategy:**

Software integration testing is the incremental integration checking out of two otherwise further included software gears on top of a solo stage to fabricate failure induced with the aid of interface defects. The project of the mixing scan is to check that components or program applications, e.g. Components in a program approach or œ one step up œ software purposes at the company degree interact without error.

Test Results:

All of the scan circumstances recounted above passed efficiently. No defects encountered.

**Acceptance testing:**

User Acceptance testing trying out is a crucial section of any mission and requires enormous participation by the tip user. It additionally ensures that the procedure meets the functional specifications.

Test Results: The entire test cases recounted above passed effectively. No defects Encountered.

13

ContentNotifier

**Chapter-6**

**CONCLUSION**

To overcome the problem of manual checking of data in websites the idea of “**ContentNotifier”** comes into play.This Application will help the people who want to know the presence of data when it is placed in the website.And also, it decreases the time to check for data. It makes it easy for you to know about the data required.

**Chapter-7**

**FUTURE ENHANCEMENTS**

This Project will include several implementations like “Required Data Analysis” which includes checking whether the data entered by the user is valid text or not and it will also focus on “Semantic Matching of Required Data” which will match data along with the semantic meaning.

**Chapter-8  
   
 APPENDIX REFERENCES**

● <https://www.google.com/>

14