Web-Monitoring Tool

Project Report

Madanayaka B. P. W it20208776

Introduction

The number of companies using internet as their business platform has increased at an exponential rate in the past decade. Those business-critical web applications need to be monitored in an objective manner to ensure optimal performance. So as a solution in this project we have created an online tool for monitor websites. Website monitoring tool helps to maintain uninterrupted monitoring access. So, we have created a tool that optimize performance, and functionality to ensure that the site is online and running smoothly 24X7. And we will also be conducting a notification service via emails, calls and text messages to notify users regarding the changes happened within the websites. This tool will give the users the ability to monitor multiple websites at the same time.

Our website monitor capabilities include with,

- Simple monitoring to prevent downtime.
- Website content monitoring.

Under simple monitoring to prevent downtime our tool will monitor the website at regular intervals (as low as 1 minute), and the user will get notified instantly when the website goes down or if the page loading time increases.

For the website content monitoring our tool auto-crawl and monitor websites to avoid unauthorized alterations or website defacement.

Importance

Website monitoring is the constant examination of its status and capabilities to ensure optimal function. In addition, the information gained from website monitoring could be used to observe trends, contributing to future planning. The process of Website monitoring is testing and verifying that end-users can interact with a website or web application as expected. Website monitoring is often used by businesses to ensure website uptime, performance, and functionality are as expected. So, with the help of our tool, users can achieve all of these possible outcomes. Most importantly our monitoring tool helps the users to increase the accountability of their websites because our tool helps identify and diagnose issues early, which can improve the quality of user experience. This can also be gained with the help of our notification system.

To use the few website monitoring tools available on the market, users need to pay a huge amount of money to get the service. But our tool is open source, and anyone can access and use the service because of the user-friendliness given by our tool. Also, in our tool users can insert any number of websites to get a real time monitoring.

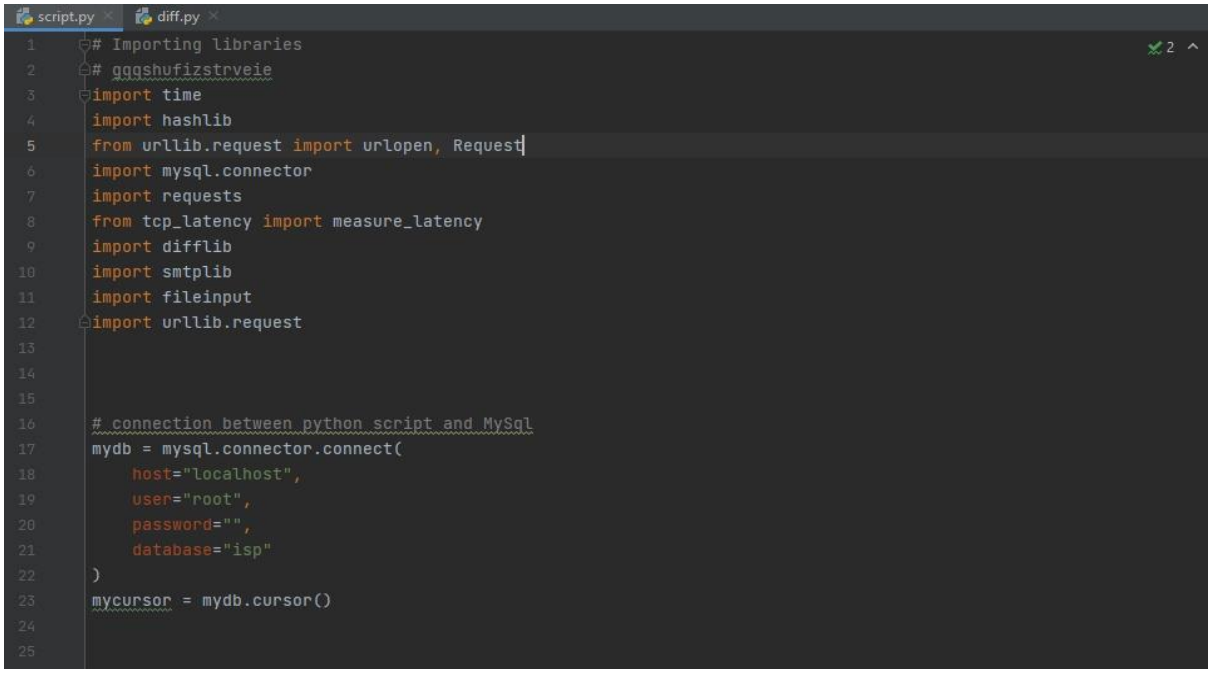
Literature

We have used python language as the main language to create this monitoring tool. As for the techniques to use within this project,

We are going to get the hash of the website from time to time and then we will be comparing those two for check any changes done with the script. In order to get the hash, we are using hashing functions and the **hashlib** library. It is a python module which is used as an interface for hashing messages easily. And we are also using SHA224 hashing algorithm.

In order to wait for a specified time period to get the new hash of the website we are going to use time python library.

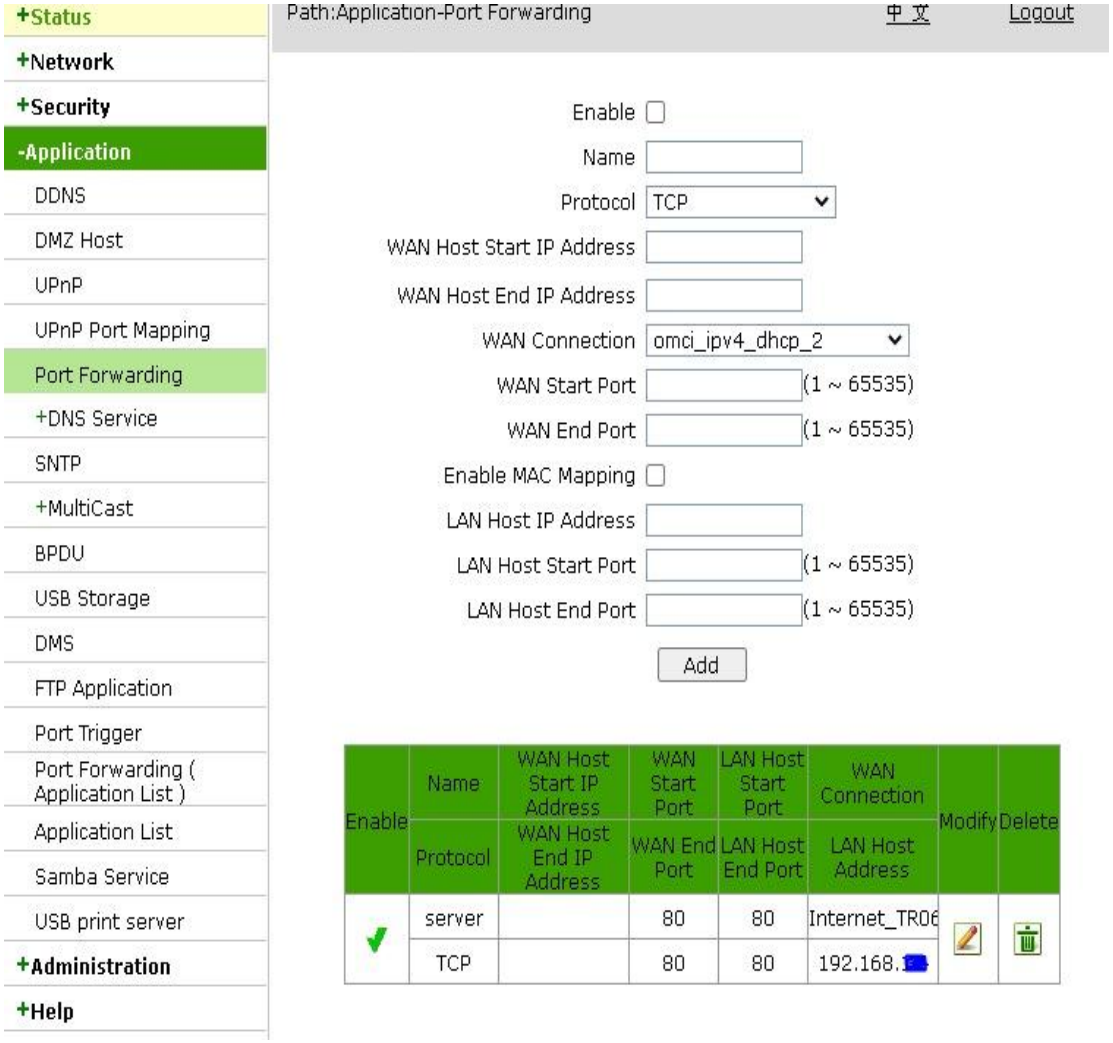
To perform the get request and load the content of the website we are using the **urllib** library.



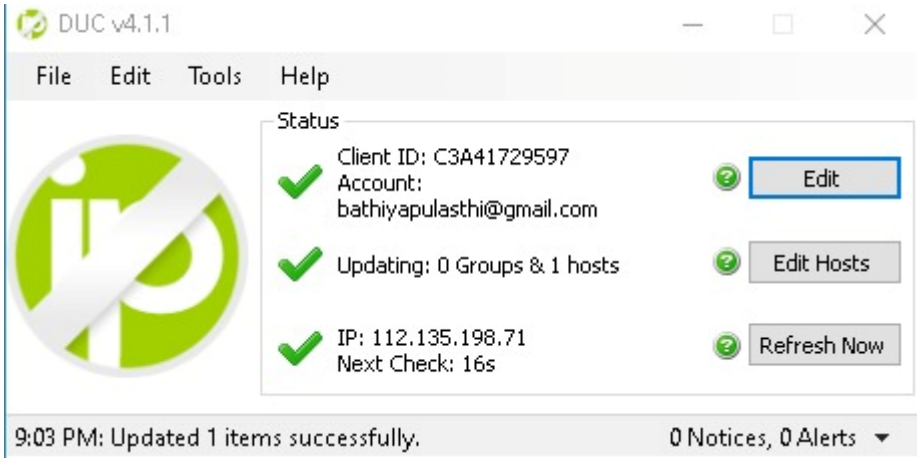
Router configurations

First of all, in order to host our website, we converted our fiber router in a server. To do that following steps were taken.

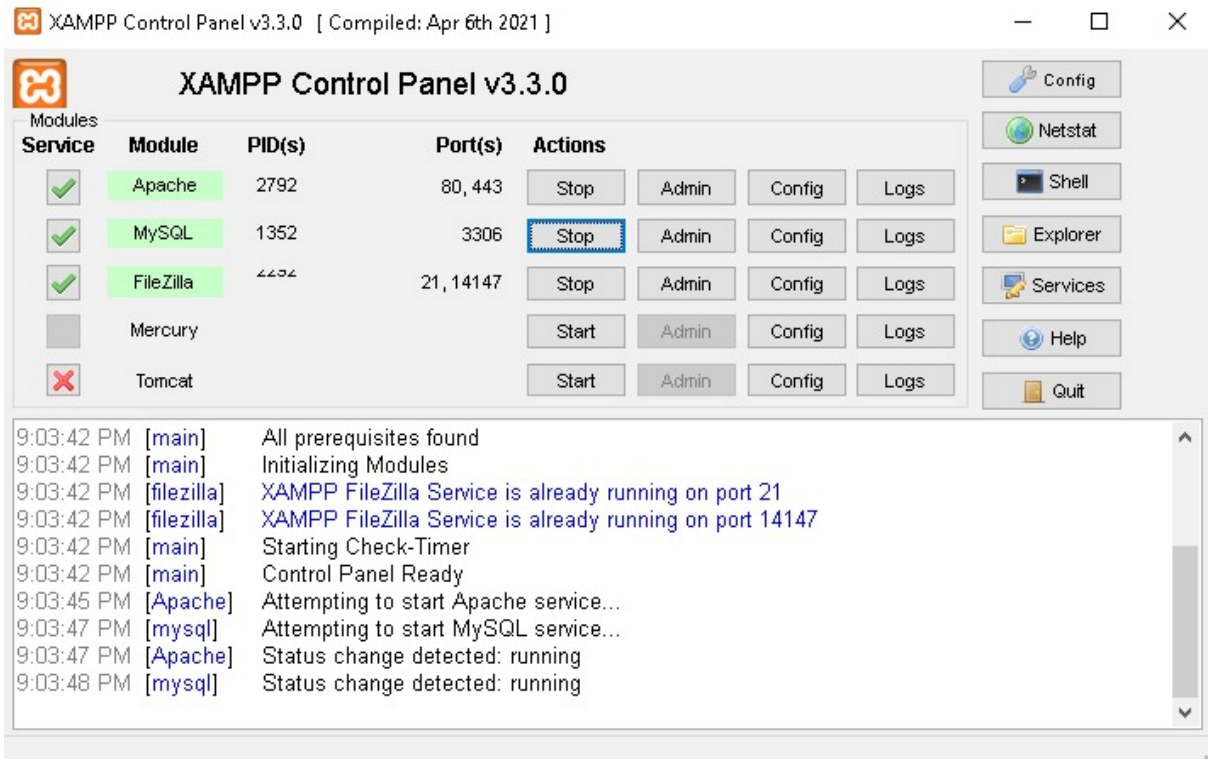
First, we enabled the port 80 from our fiber router by using port forwarding. And we set it up by inputting our laptop IP address.



By doing that anyone could access our website by using our public IP address. But it would generate a huge security concern to ourselves because that way anyone could view our public IP. So, in order to mitigate that we used a free domain name service in order to convert our public IP into a domain name. now anybody can access our website by using this URL http://sitemonitor24x7.ddns.net/.



Now that we have converted our laptop into a server. So, in order to host our website on the server we are going to use Apache and MYSQL service on XAMPP.



Functionalities

Our tool can be hosted on any Windows, Android, or iOS platform. And it gives the user the opportunity of monitoring their websites free for three months of time period.

Main purpose behind this monitoring tool is to provide a simple monitoring to prevent website downtime and to content monitoring. For that purpose, we are going to compare the website script from time to time to check any changes within the scripts. We will be doing that by getting the hash value of the website from time to time.

```
# check if new hash is same as the previous hash
if newHash == currentHash:
    continue

# if something changed in the hashes
else:
    # notify
    print("something changed")

    # again read the website
    response = urlopen(url).read()

    # create a hash
    currentHash = hashlib.sha224(response).hexdigest()

    # Inserting current hash value t DB

    sql = "INSERT INTO feature (name, marked) VALUES (%s, %s)"
    val = ("If Hash was changed8", currentHash)
    mycursor.execute(sql, val)
    mydb.commit()
    print(mycursor.rowcount, "record inserted.")
```

As you can see from the above image inside the while loop the hash values are compared from time to time to check whether the previous hash is a matched or not to the new hash.

```
#########################################################################new1-end

############################################################new2
def compare():
    # compare pre and post configurations
    show_pre = "ORIGINAL SCRIPT"
    show_post = "CHANGED SCRIPT"

    show_pre_lines = open("./initial.txt").readlines()  # convert into strings first for comparision
    time.sleep(0.5)
    show_post_lines = open("./compaired.txt").readlines()  # convert into strings first for comparision
    time.sleep(0.5)
    # four arguments required in htmlDiff function
    difference = difflib.HtmlDiff(wrapcolumn=80).make_file(show_pre_lines, show_post_lines, show_pre, show_post)
    difference_report = open("C:/xampp/htdocs/show_comparison.php", "w+")
    difference_report.write(difference)  # Writes the differences to the difference_report
    difference_report.close()
    time.sleep(0.5)
#############################################################################new2-end
def status_check():

    response = urllib.request.urlopen(url)
    status_code = response.getcode()
    if status_code == 200:
        print('UP')
```

Also, in order to alert users regarding any changes happened within the website we are maintaining an alerting system to send a notification by using emails, calls and text messages. The image below image illustrates how the notification scenario is handled by our system.

```
# Defining The Message
SUBJECT = "Your Site was changed"
TEXT = "Please take immediate action because your website has been changed. Please disregard this mess
       "\n" \
       "\n" \
       "Thanks & BR"

message = 'Subject: {}\n\n{}'.format(SUBJECT, TEXT)

# Sending the Email
smtp.sendmail("absitemonitor24x7@gmail.com", "bathiyapulasthi@gmail.com", message)

# Terminating the session
smtp.quit()
print("Email sent successfully!")

############################################################new-4end
```

Your Site was changed

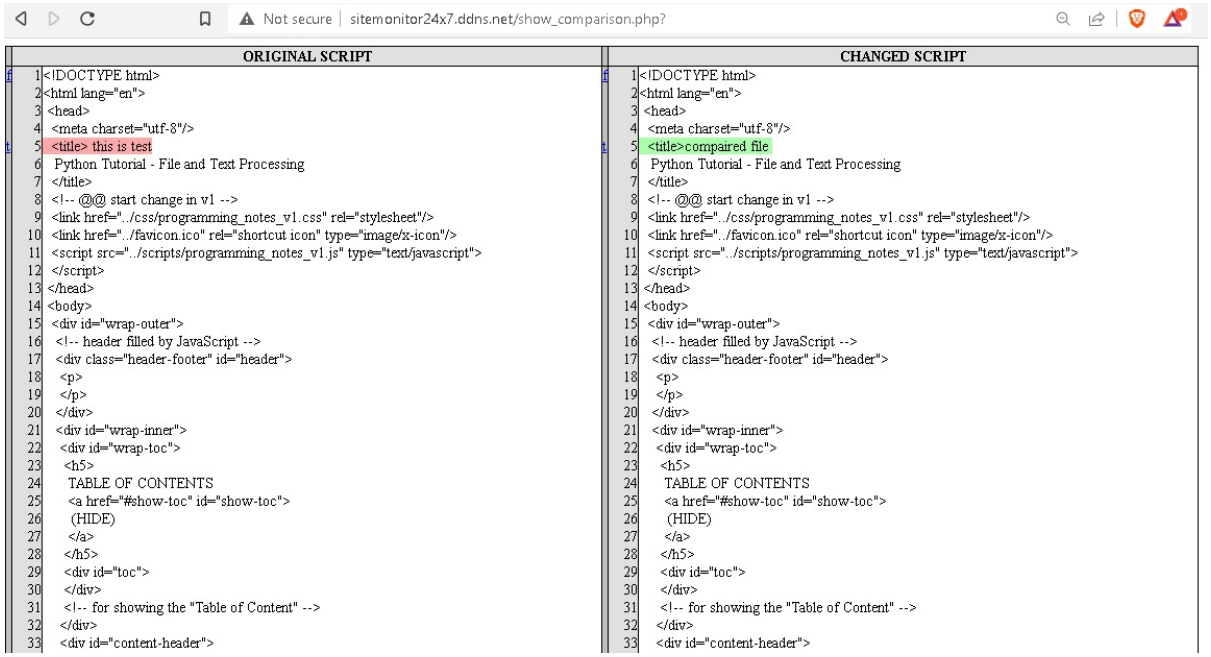
absitemonitor24x7@gmail.com
to bcc: bathiyapulasthi

Please take immediate action because your website has been changed. Please disregard this message if it is legitimate.

Thanks & BR

Reply Forward

The script comparison within our tool will be illustrated as below image.

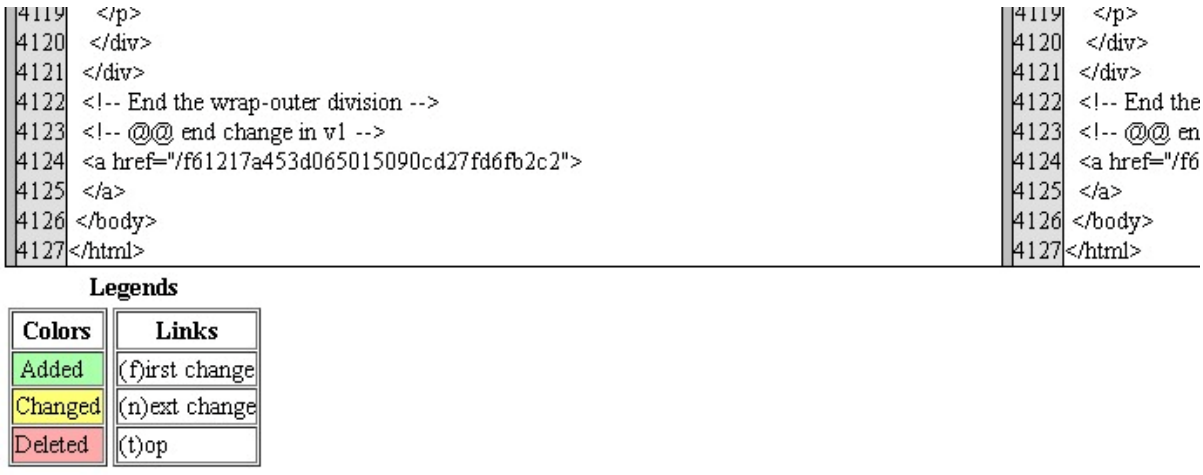


Any changes done within the website script will be shown as,

Green → Added values to the script

Yellow → Values are changed

Red → Values are deleted



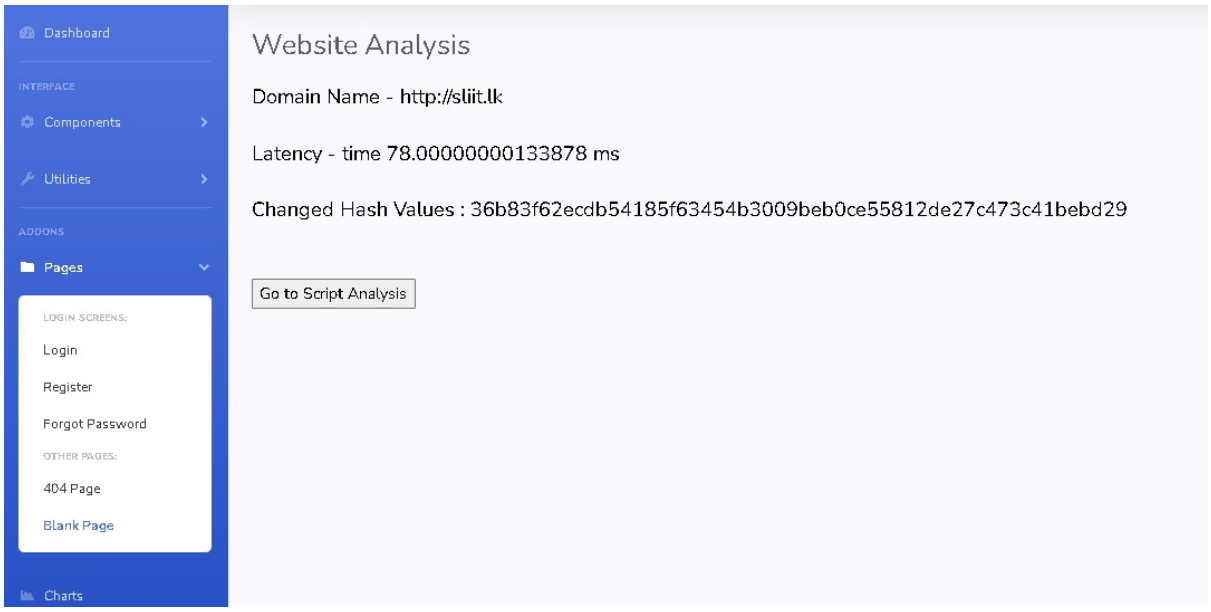
Legends

Colors	Links
Added	(f)irst change
Changed	(n)ext change
Deleted	(t)op

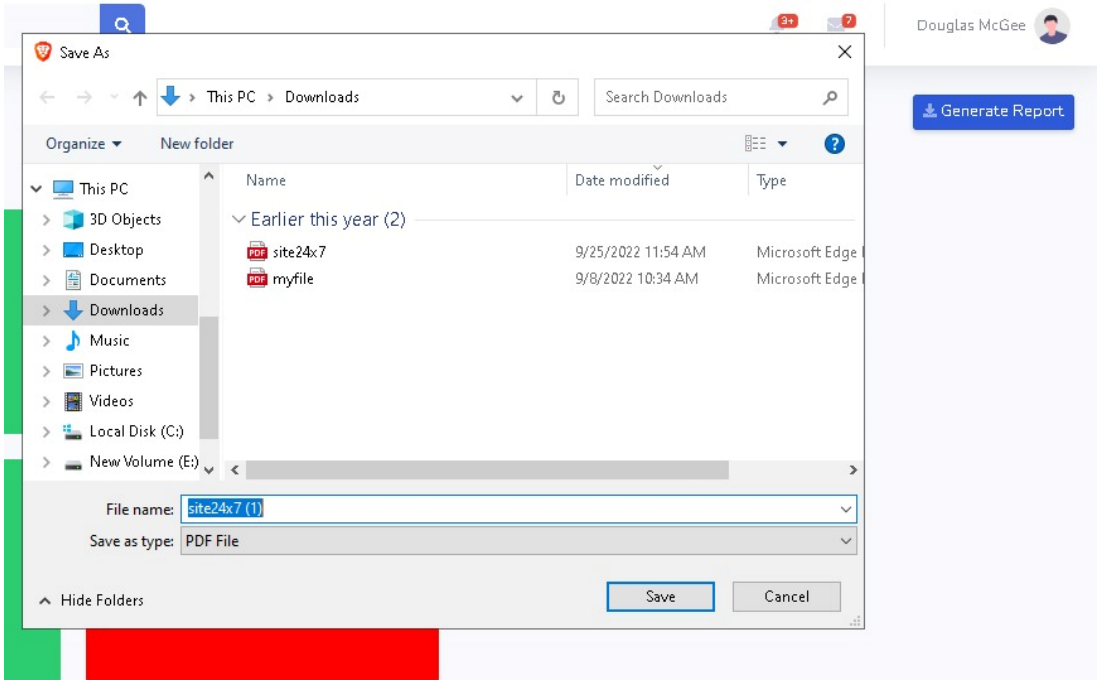
Also, within our tool, it will generate the latency of the given website. We will get the latency of the website by using the below command.

```
# get this latency and Display
latency=measure_latency(host='bathiyapulasthi.wixsite.com')
print (latency)
```

And it will be illustrated within our website as below.

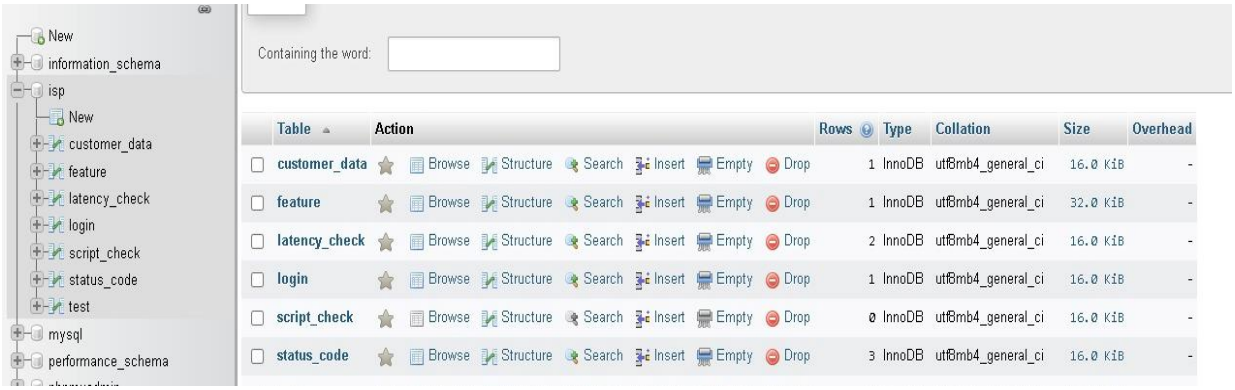


If the user needs to generate a report by including all this information, it can also be done via our tool.



Database

Now let’s move to the database of our website.

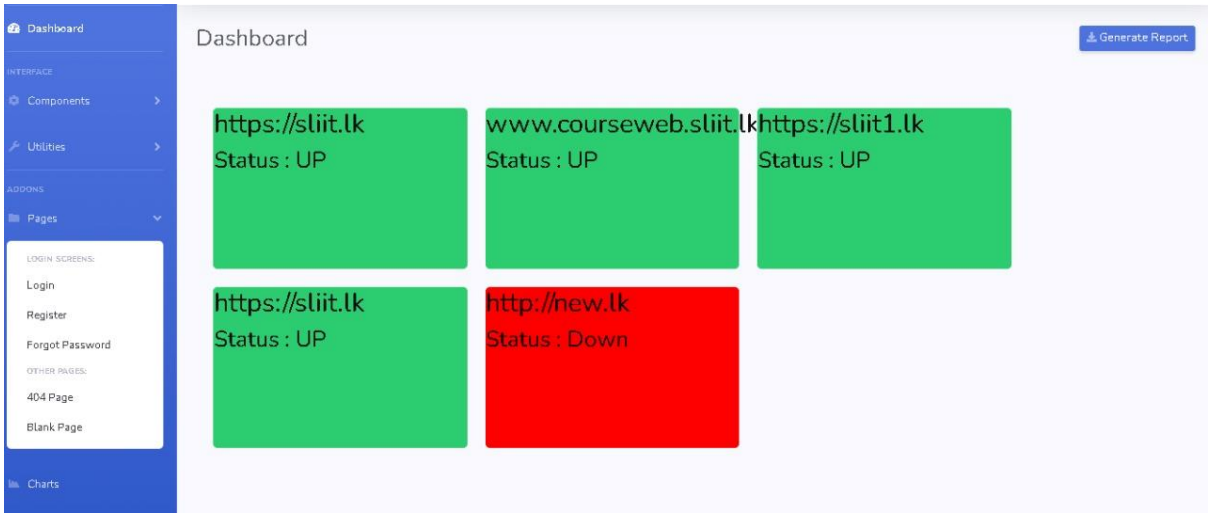


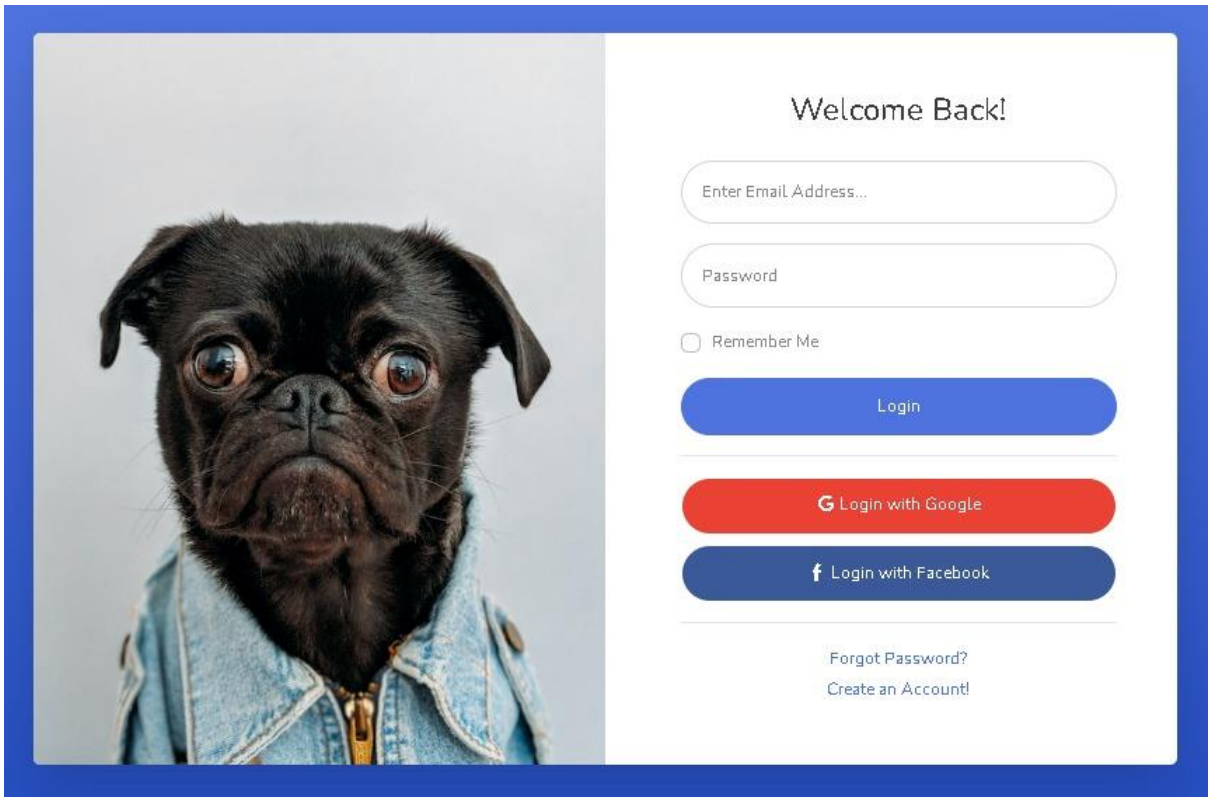
Above image shows our database. Through the database the web interface will communicate with the python script.

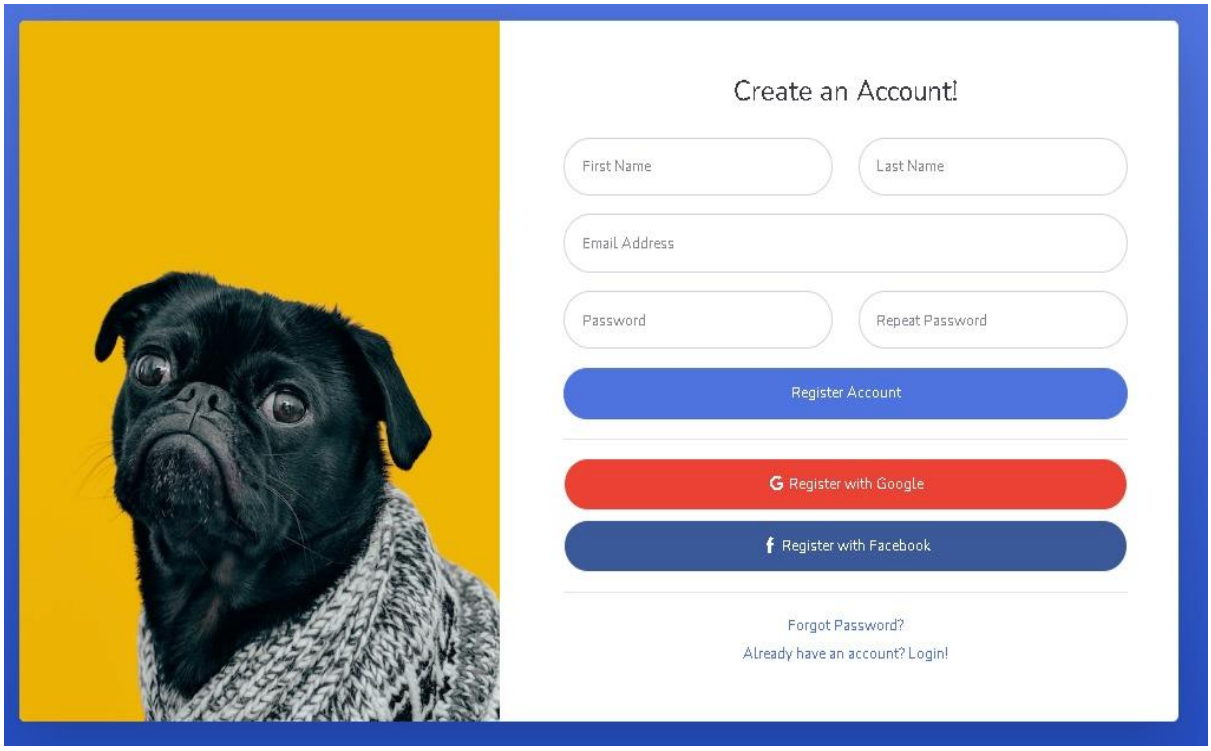
All the necessary data will be taken via the database. Web interface will use below method in order to import customer inputted data into the database.

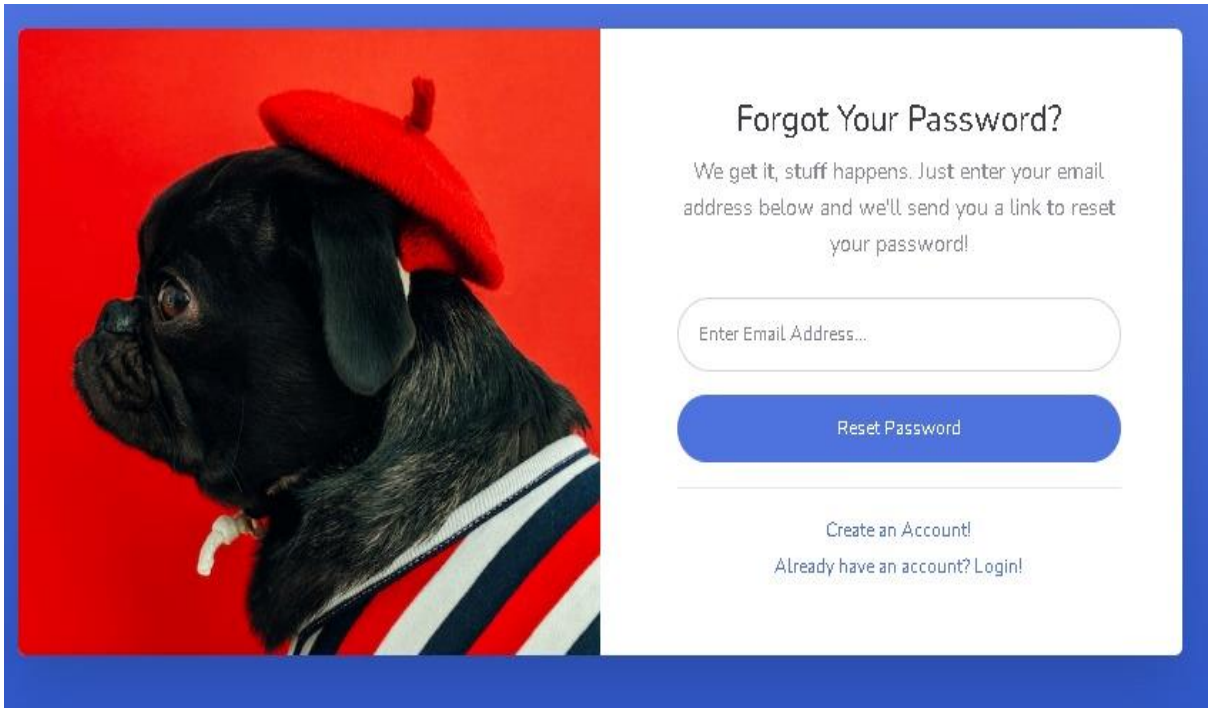
```php

<?php

    include_once 'db_connect.php'
?>

<?php

    $DomainURL=$_POST["dname"];
    $email=$_POST["email"];
    $telephone=$_POST["phone"];


    $sql="insert into customer_data(C_ID,C_URL,C_Email,C_Phone)values('','$DomainURL','$email','$telephone')";

     if(mysqli_query($conn,$sql)){
         echo "<script>alert ('Your request made')</script>";
         header("Location:/index.php");
     }
         else{
             echo "<script>alert ('Your request Unsuccesful Please try again')</script>";
     }

     mysqli_close($conn);
?>

```

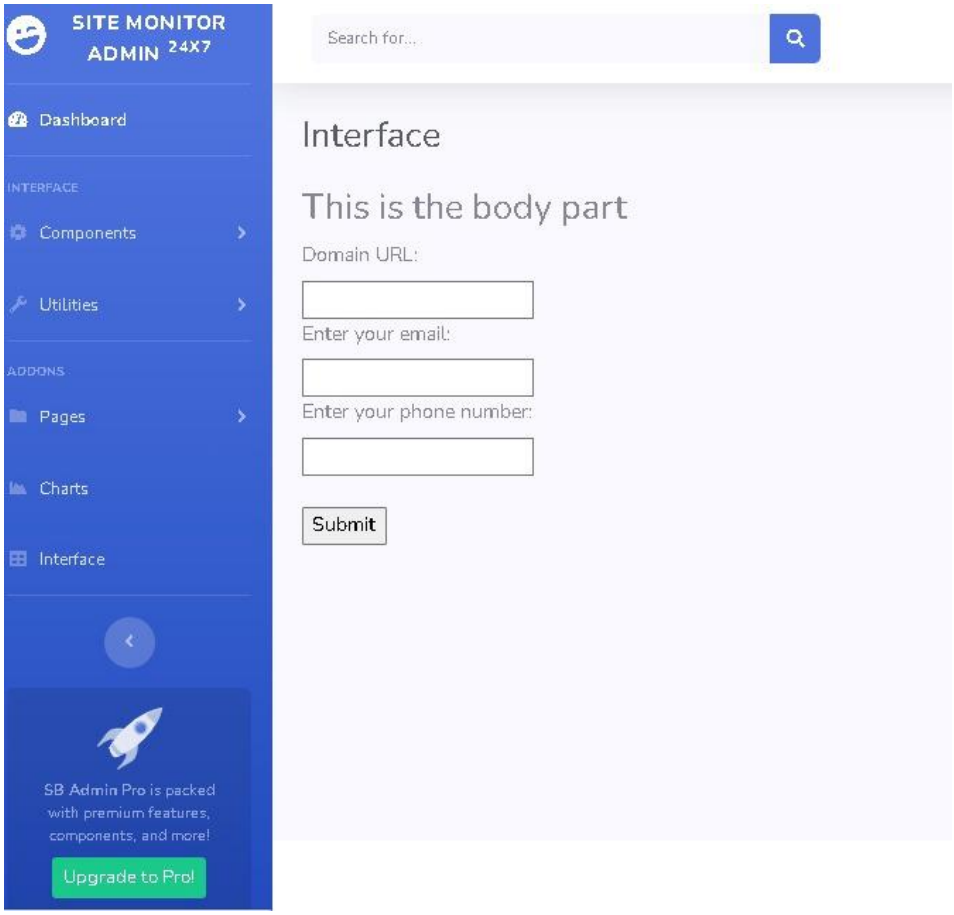
Now for the interface. Our site contains mainly five pages. Login page, register page, forgot password page, Dashboard and interface.






Create an Account!
First Name
Last Name
Email Address
Password
Repeat Password
Register Account
Register with Google
Register with Facebook
Forgot Password?
Already have an account? Login!


Forgot Your Password?
We get it, stuff happens. Just enter your email address below and we'll send you a link to reset your password!
Enter Email Address...
Reset Password
Create an Account!
Already have an account? Login!



All the necessary data required from the customer will be inputted into the interface page.

All of these functionalities will be up and running 24X7 within our script from our previously configured server.

Methodology and Business model

We have agile methodology to finish tasks of the project within the timeline of the SCRUM period with the help of Jira software.

We got the scratch of our web application tool by using mockflow. With the help of the scratch, we get a better understanding regarding the requirements and functionalities of our web application.

Once we got the functionalities, we are going to prioritize them according to their importance.

We recognized the main function as the script change identification function after we prioritization.

As I explained earlier, we used the following approach to implement this function.

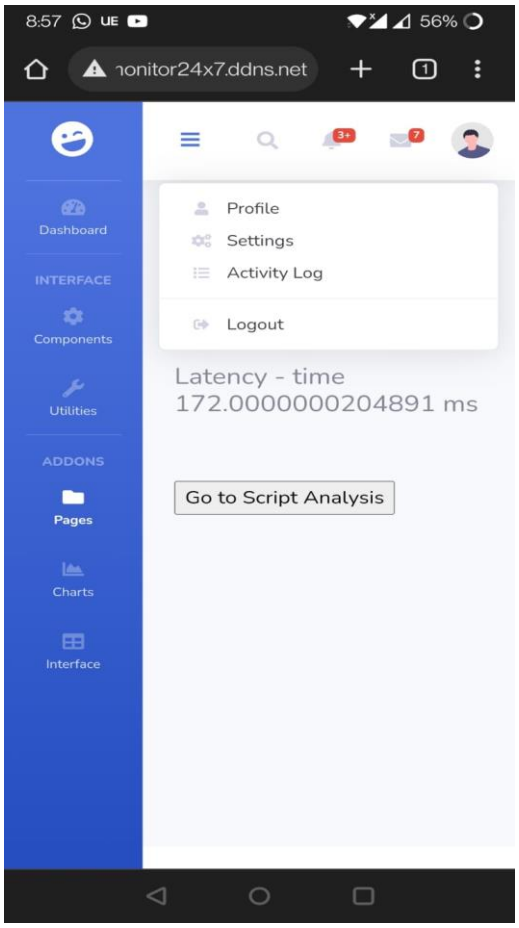
- Examine the URL that the user is wished to examine.
- Hash the entire website.
- Wait for the specified number of seconds.
- If there are any changes from the previous hash, we going to proceed with the alerting system; otherwise, wait and then take the hash again.

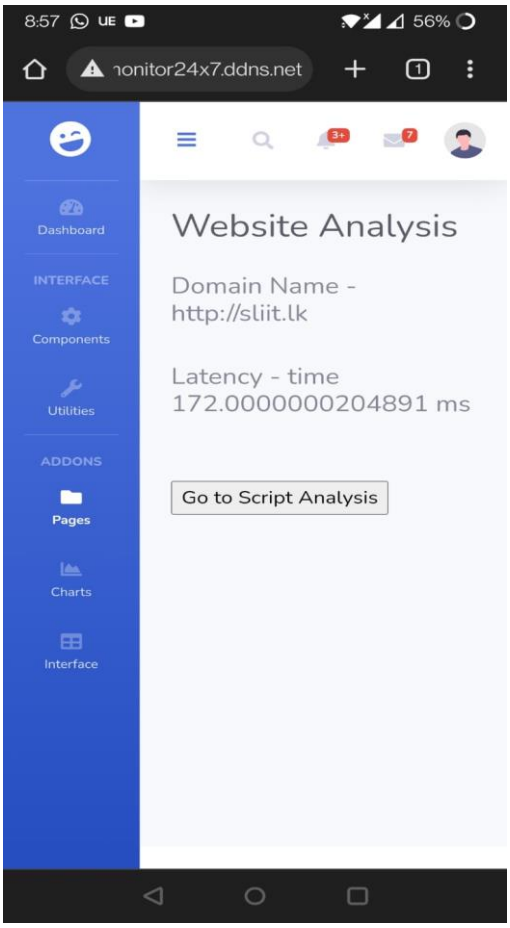
After the main goal of the project is completed, we are going to continue with the other functionalities such as “sitedowntime () function” and the alerting system.

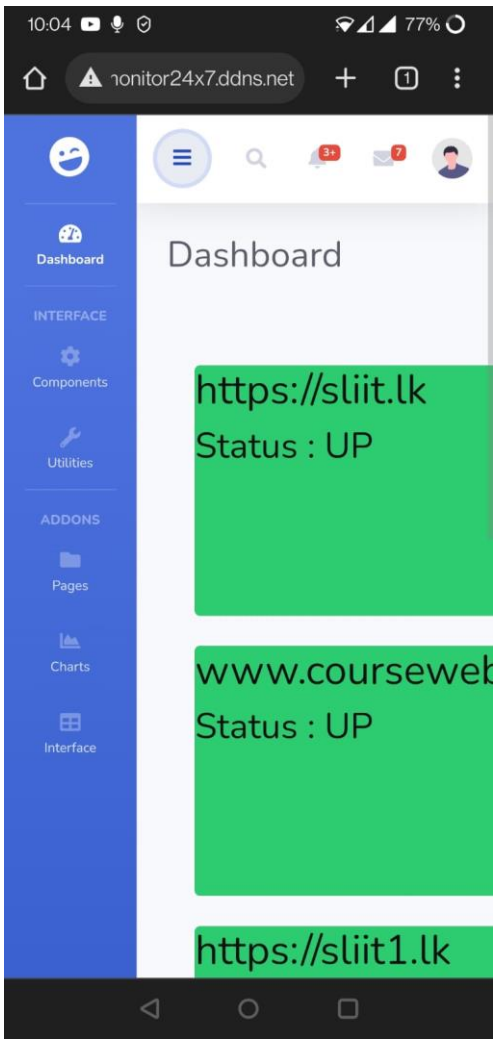
We are going to host our web application with the database on a hosting service.

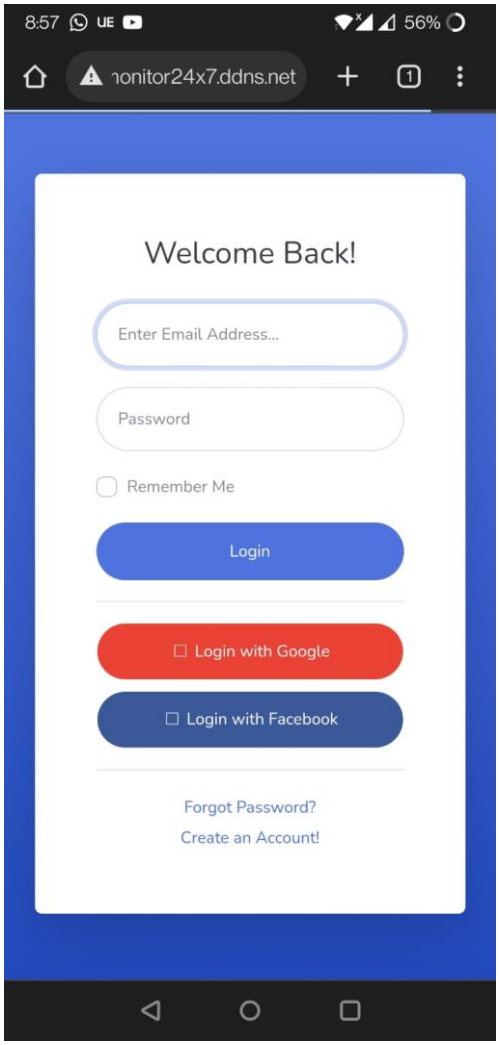
In the final SCRUM period, we tested and reviewed all the possible use case scenarios for our product.

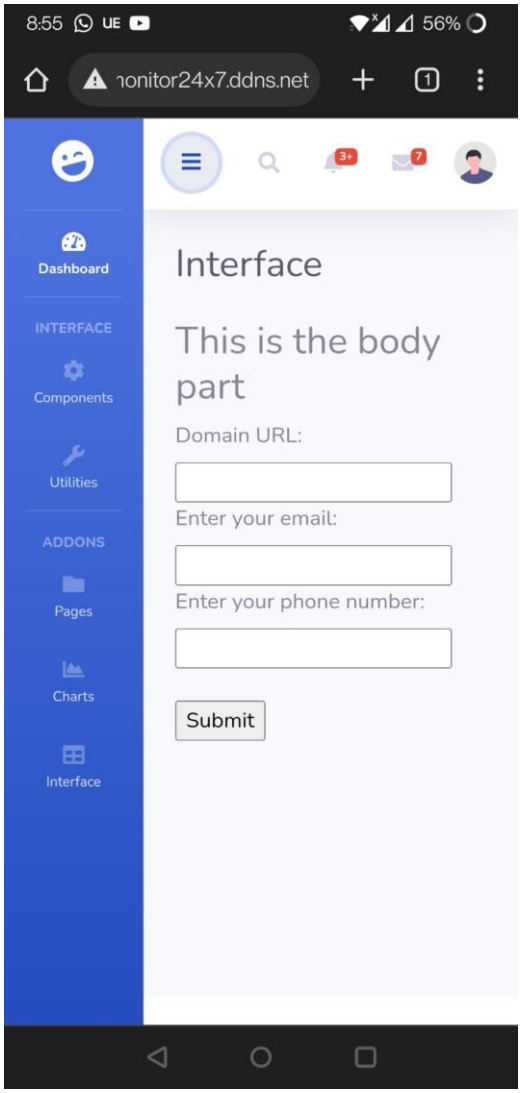
Mobile interface of the site






10:04
77%
monitor24x7.ddns.net
Dashboard
INTERFACE
Components
Utilities
ADDONS
Pages
Charts
Interface
Dashboard
https://sliit.lk
Status : UP
www.courseweb
Status : UP
https://sliit1.lk


8:57
UE
56%
monitor24x7.ddns.net
Welcome Back!
Enter Email Address...
Password
Remember Me
Login
Login with Google
Login with Facebook
Forgot Password?
Create an Account!


8:55
56%
onitor24x7.ddns.net
Dashboard
INTERFACE
Components
Utilities
ADDONS
Pages
Charts
Interface
Interface
This is the body part
Domain URL:
Enter your email:
Enter your phone number:
Submit


8:57
56%
onitor24x7.ddns.net

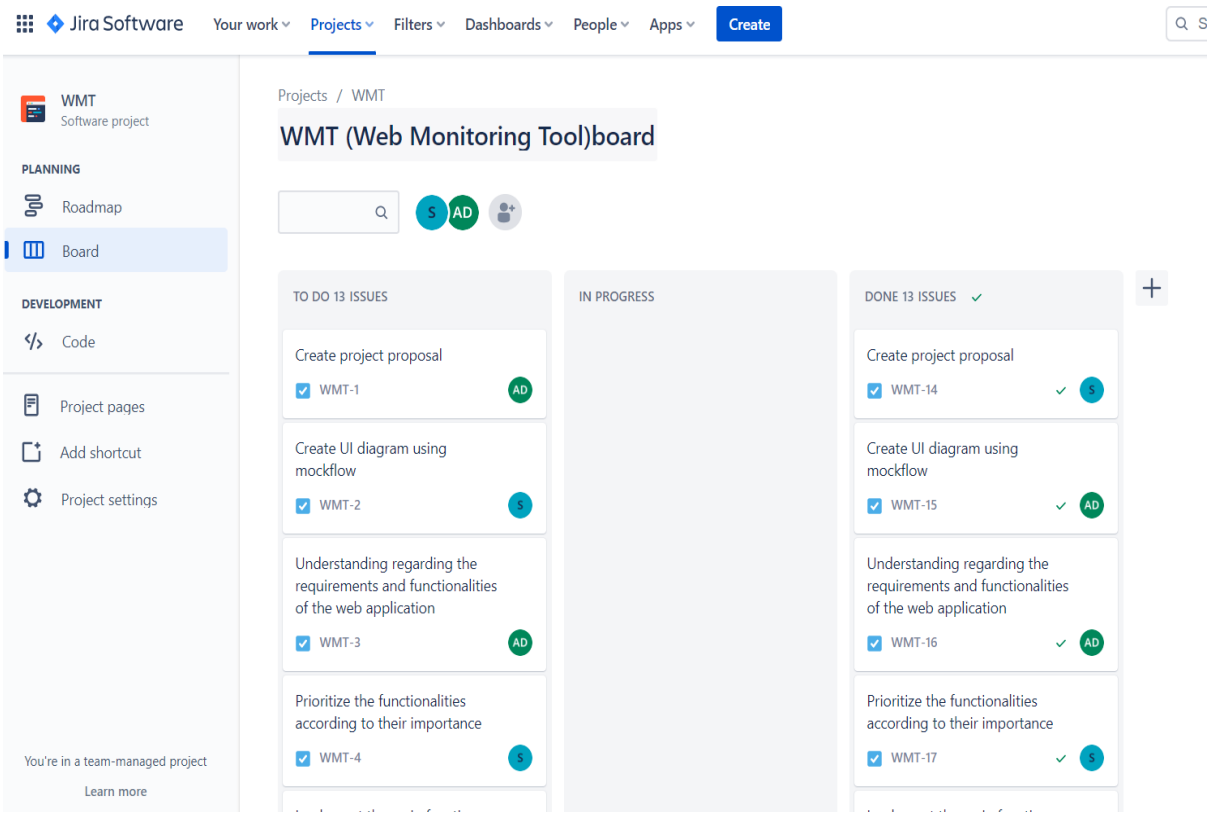
Technology and architecture

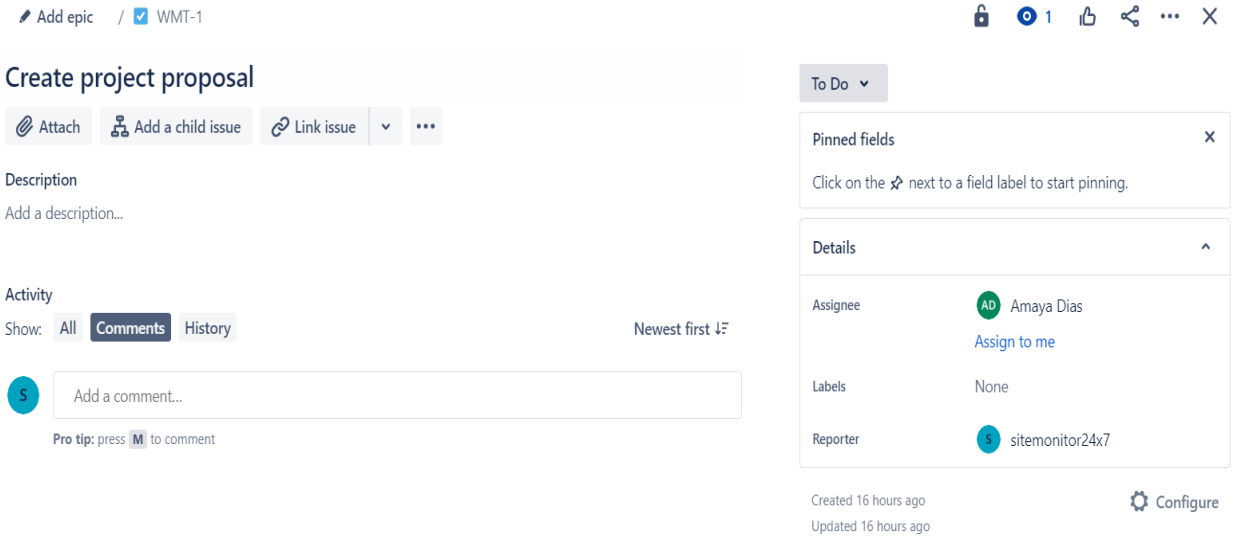
Some of the technologies and software that we used in order to conduct this project are visual studio, Jira software, mockflow, XAMMP server and pager duty.

For the hosting part we are going to use resources from cloud providers.

Timeline with agile sprints

1. Create project proposal
2. Create UI diagram using mockflow
3. Understanding regarding the requirements and functionalities of the web application
4. Prioritize the functionalities according to their importance
5. Implement the main function - script change identification function
6. Implement site downtime function
7. Implement alerting function
8. Implement other necessary function
9. Create the web application interface
10. Plug in the database
11. Testing phase - Test all the possible use case scenarios
12. Host the web application
13. Final review period – monitoring


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
WMT
Software project
PLANNING
Roadmap
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / WMT
WMT (Web Monitoring Tool)board
TO DO 13 ISSUES
IN PROGRESS
DONE 13 ISSUES
Create project proposal
WMT-1
Create UI diagram using mockflow
WMT-2
Understanding regarding the requirements and functionalities of the web application
WMT-3
Prioritize the functionalities according to their importance
WMT-4
Create project proposal
WMT-14
Create UI diagram using mockflow
WMT-15
Understanding regarding the requirements and functionalities of the web application
WMT-16
Prioritize the functionalities according to their importance
WMT-17


Add epic
WMT-1
Create project proposal
Attach
Add a child issue
Link issue
Description
Add a description...
Activity
Show:
All
Comments
History
Newest first
Add a comment...
Pro tip: press M to comment
To Do
Pinned fields
Click on the next to a field label to start pinning.
Details
Assignee
Amaya Dias
Assign to me
Labels
None
Reporter
sitemonitor24x7
Created 16 hours ago
Updated 16 hours ago
Configure

References

[1] [Online]. Available: https://www.uptrends.com/what-is/website-monitoring#:~:text=Website%20Monitoring%20is%20an%20all,use%20the%20site%20as%20expected..

[2] "pagerduty," [Online]. Available: https://www.pagerduty.com/. [Accessed 20 8 2022].

[3] B. M. A. S. Sandrine Vaton, "Web View: Measuring & Monitoring Representative Information on Websites," 11 April 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8685876. [Accessed 20 08 2022].

[4] L. A. Bala Simon, "Website Monitoring," taylorfranci, 2021. [Online]. Available: https://www.taylorfrancis.com/chapters/edit/10.1201/9781003203131-28/website-monitoring-raj-bala-simon-laxmi-ahuja. [Accessed 22 08 2022].

[5] ATLASSIAN, "jira Software," ATLASSIAN, [Online]. Available: https://www.atlassian.com/software/jira?&aceid=&adposition=&adgroup=95003645649&campaign=9124878702&creative=542638212653&device=c&keyword=jira%20software&matchtype=e&network=g&placement=&ds_kids=p51242189321&ds_e=GOOGLE&ds_eid=700000001558501&ds_e1=GOOGL. [Accessed 22 08 2022].