

Имена: Христо Спасов

ФН: 62278

Имена: Таня Желева

ФН: 62288

Начална година: 2018

Програма: бакалавър, (СИ)

Курс: 3

Тема: Уеб архив (wayback machine)

Дата: 2021-06-20

Предмет: w16prj_SI_final

имейл: hristo.b.spasov@gmail.com

имейл: tanya.n.zheleva@gmail.com

преподавател: доц. д-р Милен Петров

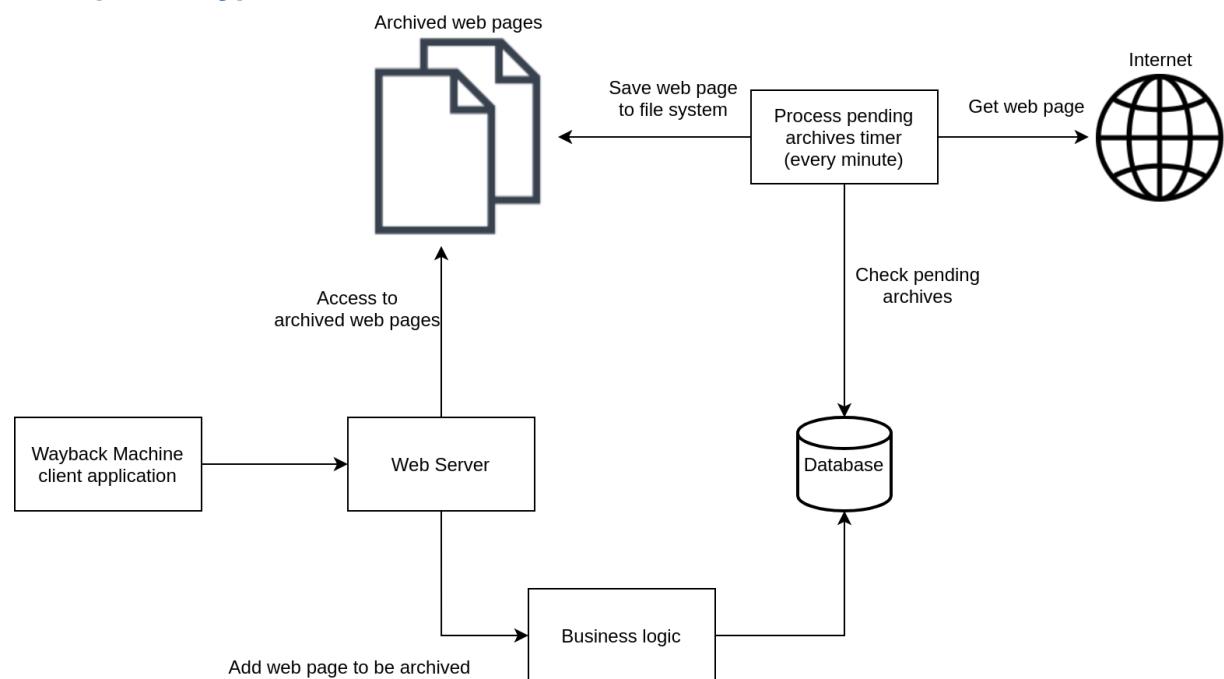
ТЕМА: Уеб архив (wayback machine)

1. Условие

Да се направи система, подобна на „<https://archive.org/web/>“, която по зададен URL прави архив на дадена страница. След архивиране може да се влезе и да се навигира по страници и да се търси архив за страница. Ако има - всяко от архивираните се показва в календар или списък.

2. Въведение

2.1. Архитектура



фиг. 1 - Архитектура на системата Wayback Machine

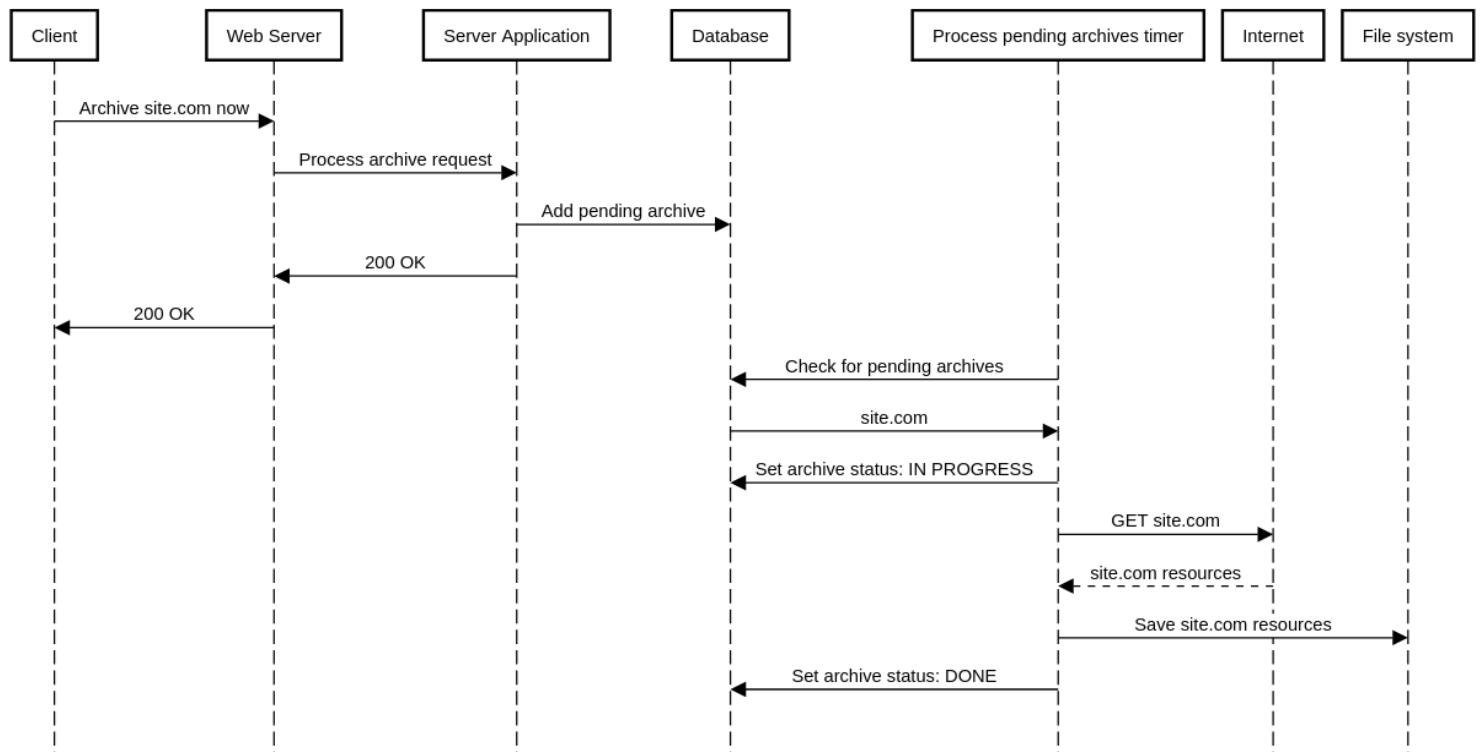
Wayback machine се състои от следните компоненти:

- Wayback Machine client application - уеб страница, която може да се отваря от браузър. Чрез нея потребителите задават кои страници да се архивират или чрез нея може да се визуализира избрана архивирана страница.

- Web Server - уеб сървър, който приема заявки за архивиране на страници и заявки за достъп до вече архивирани страници.
- Business logic - сървърно приложение, което записва в базата от данни заявки за архивиране на страници, записва статус на изпълнение на заявките, предоставя метаданни за вече архивирани сайтове
- Database - база от данни, в която се записва информация за архивирани сайтове. Тя представлява индекс за наличните архиви и също така съхранява техните статуси. Базата данни не съдържа самите архивирани страници.
- Archived web pages - архив от страници, записани на файлова система
- Process pending archives timer - скрипт, който се изпълнява всяка минута, проверява за направени заявки за архивиране и ги обработва като изтегля страниците и обновява статуса на заявката в базата данни.

2.2. Процеси

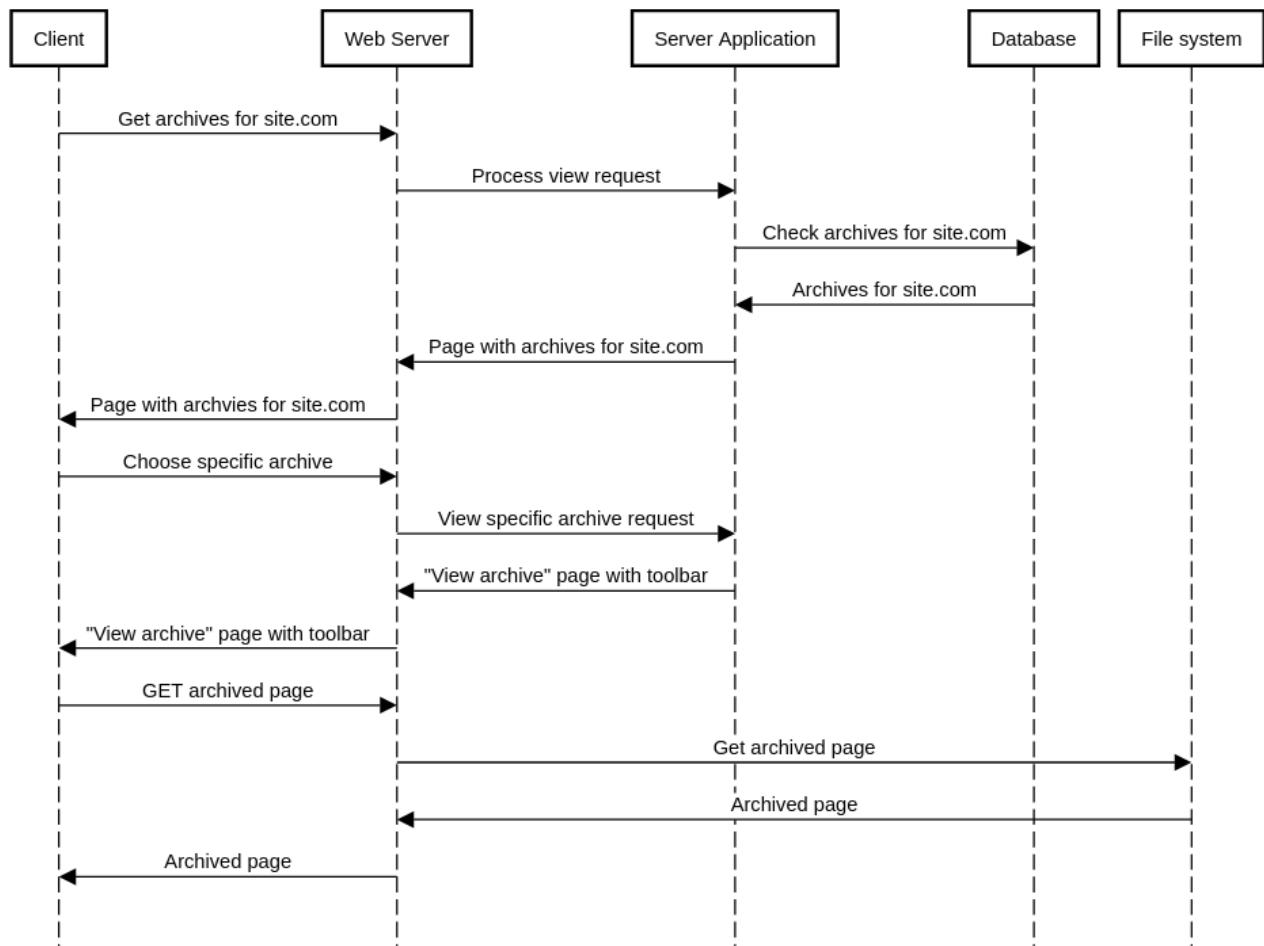
2.2.1. Архивиране на сайт



фиг. 2 - Диаграма на последователностите при архивиране на сайт

Клиентът изпраща заявка за архивиране, подавайки URL на сайт, който да се архивира. Заявката се записва в базата и клиентът получава отговор, че заявката е изпълнена успешно. След няколко секунди се активира скрипта за обработка на заявки за архивиране. Той открива, че има заявка, която се нуждае от обработка. Задава ѝ статус „in progress“, след което изтегля ресурсите за сайта. Накрая задава статус на заявката „done“.

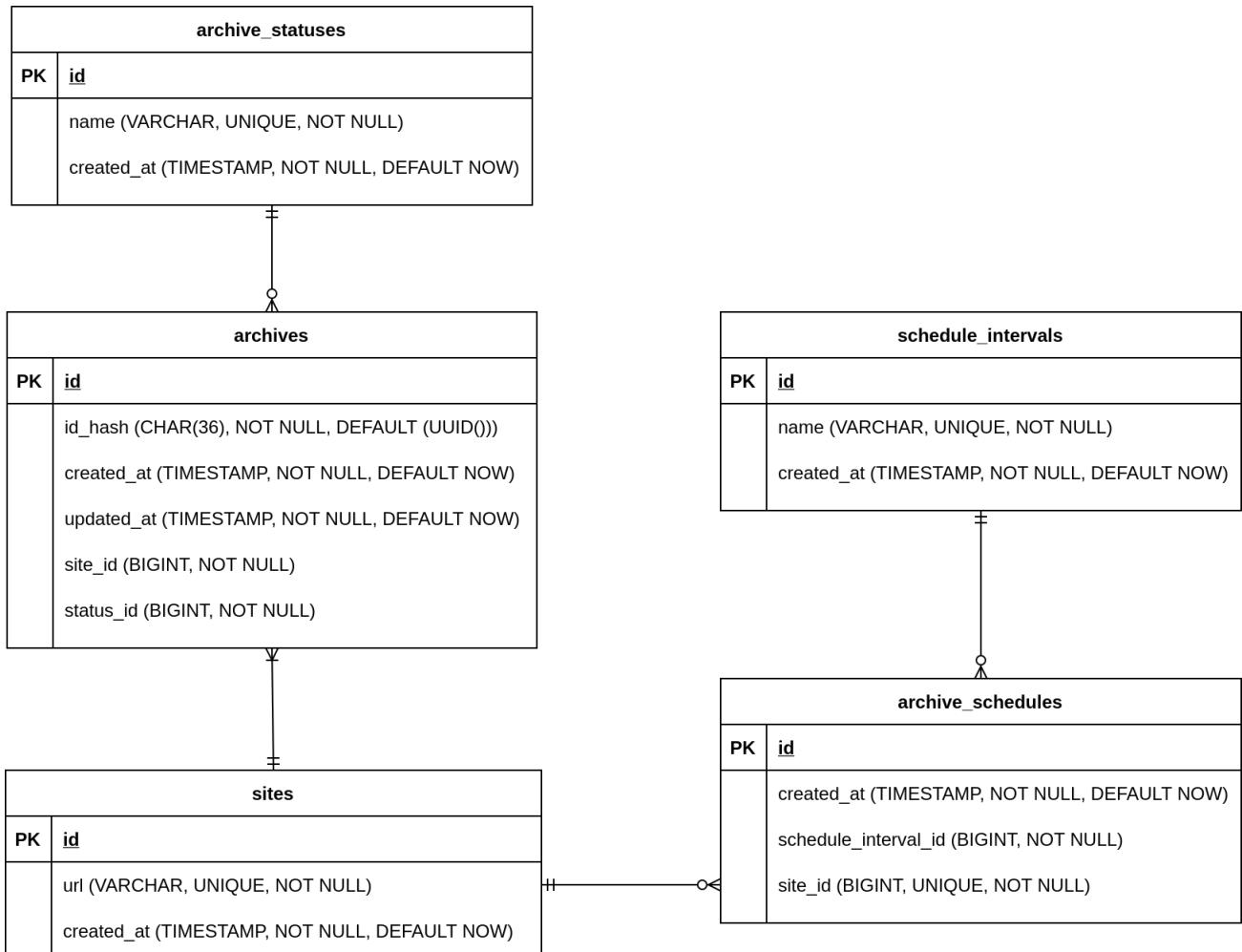
2.2.2. Визуализация на архивиран сайт



фиг. 3 - Диаграма на последователностите при визуализация на сайт

Клиентът изпраща заявка за преглед на архивиран сайт. Сървърното приложение проверява в базата данни за наличие на архиви за този сайт и връща на клиента резултата. Клиентът избира архив от конкретен момент и сървърът връща страница с лента, показваща URL на текущ сайт и опции за избор на друг момент от време. Под лентата е разположен iframe с точен адрес на ресурсите за сайта. Клиентът изпраща заявка за вземане на ресурсите от сайта. Уеб сървърът ги прочита от файловата система и ги предоставя на клиента.

2.3. База от данни



фиг. 4 - Диаграма на класовете на базата данни

Описание:

- **sites** - в нея се записват всички URL-и на сайтове.
- **archives** - в нея се записват всички архиви за даден сайт, техните timestamp-и, както и статус на архива - дали е pending, in progress или done. Колона **id_hash** се използва за намиране на ресурсите за сайта върху файловата система. Ресурсите за даден сайт се записват в директория с име стойността на **id_hash**.
- **archive_statuses** - номенклатурна таблица. Съдържа гореспоменатите три статуса.
- **archive_schedules** - съдържа информация даден сайт с какъв график на периодично архивиране е.
- **schedule_intervals** - номенклатурна таблица. Съдържа възможните периоди на архивиране: всеки месец, веднъж на всеки 6 месеца, всяка година, веднъж на 3 години, веднъж на 5 години и веднъж на 10 години.

За всички първични и външни ключове, както и колони, по които се извършва търсене, са създадени индекси. Създаден е тригер за автоматично обновяване на колона `archives.updated_at` при юпдейт в таблицата.

3. Теория

3.1. Архивиране на сайт

Отварянето на една интернет страница обикновено се състои от повече от една заявка, тъй като страниците включват в себе си и други ресурси, например изображения, стилизация и скриптове. Необходимо е да се осъществи изтеглянето на всички необходими ресурси по начин, който да позволи бъдещата визуализация на страницата по начин, по който тя първоначално е изглеждала. Командата `wget` предоставя тази възможност и системата `wayback machine` се възползва от следните аргументи на `wget`:

- `--directory-prefix` - задаване на място във файловата система, където ресурсите за страницата да се запазят. Тази опция позволява мястото за съхранение да се контролира от сървърното приложение и така може да се осъществи бъдещо намиране на вече архивирани сайтове.
- `--no-cookies` - забранява използването на бисквитки. Тази опция гарантира консистентността на работата на скрипта за архивиране на сайтове.
- `--page-requisites` - тази опция задава да се изтеглят всички ресурси, които са необходими за правилната визуализация на една страница. Ресурсите се запазват в същата йерархия, каквато е от гледната точка на един браузър.
- `--span-hosts` - тази опция задава да се изтеглят ресурси и от хостове, различни от първоначално заявения. Тази опция е необходимо допълнение към „`--page-requisites`“. Без нея се изтеглят ресурси само от първоначално заявения хост.
- `--convert-links` - конвертира връзките към ресурси в свалените страници по такъв начин, че да сочат към локално свалените ресурси

3.2. Визуализация на архивиран сайт

За да се визуализира една архивирана страница е нужно да се подаде пълния URL. Той се приема и се прави заявка към базата данни за всички налични timestamps за архиви, които отговарят на подадения URL. След това те се визуализират и е нужно да се избере един от тях. При избор на някой от изброените timestamps, се отваря страницата от архива, който се намира на конкретен път във файловата система. Самият път до архива се определя динамично в зависимост от подадения URL, избрания timestamp и други параметри, които се взимат от базата данни.

3.3. Съхранение на архивирани сайтове

За съхранение на информация за запазени сайтове се използва базата от данни. Самите архивирани сайтове се съхраняват в конкретна публична директория. Връзката между запис в базата от данни и архив се осъществява чрез генериран UUID: в базата от данни за даден архив се записва в колона `id_hash` генеририаният UUID, а след това в директорията, където се съхраняват архивите, се създава нова директория с име генеририаният UUID, и в новата директория се изтеглят данните за архива.

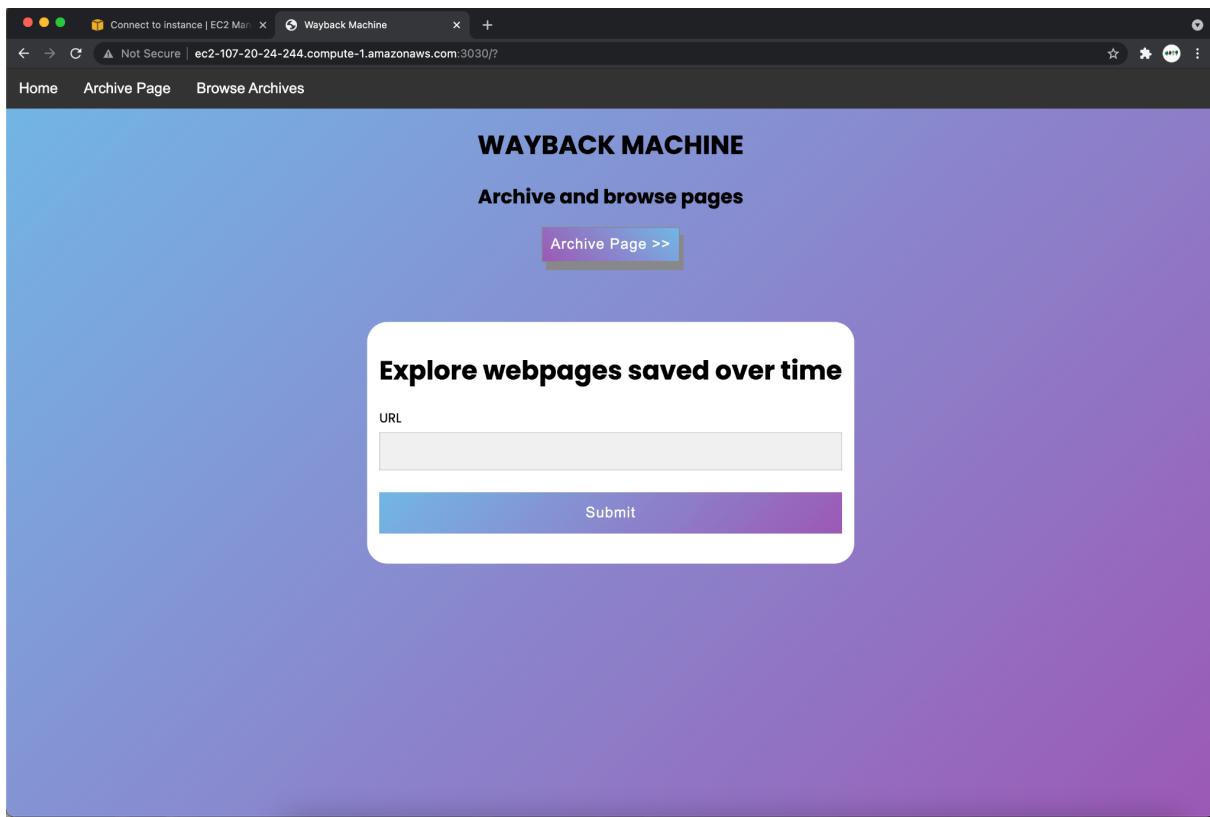
4. Използвани технологии

- Системата е проектирана да работи на операционни системи **Ubuntu 20.04** и **Windows 10**.
- За обработка на заявки се използва уеб сървър **Apache2**.
- Периодичното изпълнение на скриптове се осъществява от **systemd timer** или **Task Scheduler**.
- Базата от данни е **MySQL**.
- Сървърната логика е изградена с езика **PHP**
- За изграждане на фронт енд са използвани технологиите **HTML**, **CSS** и **JavaScript**

5. Инсталация и настройки

5.1. Инсталация на системата на Ubuntu 20.04

1. Свалете изходния код на системата за Ubuntu 20.04.
2. Отворете директория `wbmachine`.
\$ cd ./wbmachine
3. Изпълнете със супер потребител инсталационния скрипт:
\$ sudo ./setup/install.sh
4. Отворете `localhost:3030`. Ако видите началната страница на Wayback Machine, значи системата е успешно инсталирана



фиг. 5 - Начална страница на системата Wayback Machine

5.1.1. Деинсталация на системата

Можете да деинсталирате системата и да изтриете всички данни от нея със скрипта за деинсталация:

```
$ sudo ./setup/uninstall.sh
```

5.2. Инсталация на системата на Windows 10

Предварителни изисквания:

- Необходимо е да имате инсталиран уеб сървър Apache 2.4 или по-нова версия
- PHP 7.0.0 или по-нова версия
- Mysql 8.0 или по-нова версия
- Необходимо е да имате инсталиран Wget For Windows. Можете да го свалите от
<https://sourceforge.net/projects/gnuwin32/files/wget/1.11.4-1/wget-1.11.4-1-setup.exe/download>

1. Свалете изходния код на системата за Windows 10.

2. Добавете следната конфигурация към конфигурационния файл за Apache httpd-vhosts.conf

```
Listen 0.0.0.0:3030

<VirtualHost *:3030>
    SetEnv DBHOST localhost
    SetEnv DBNAME wbmachine
    SetEnv DBUSER wbmachine
    SetEnv DBPASS 'ParolataESlozhna'
    SetEnv S3_URL 'https://wbmachine.s3.amazonaws.com'
    SetEnv S3_ENABLED 0
    DocumentRoot c:/wbmachine/views/public
    AliasMatch ^/(?!js|css|sites)/(.+)$ c:/wbmachine/server/index.php
</VirtualHost>

<Directory "c:/wbmachine/">
    Require all granted
</Directory>
```

3. Добавете следният ред в php.ini файловете, които ще се ползват при изпълнение на сървъра и при изпълнение на скрипта за теглене на страници:

```
include_path="c:/wbmachine"
```

Например ако използвате WAMP, съответните версии и сте избрали същото място на инсталация, то това би трябвало да са следните файлове:

```
C:\wamp64\bin\php\php7.3.21\phpForApache.ini
C:\wamp64\bin\php\php7.3.21\php.ini
```

4. Преместете изходния код на системата в c :\wbmachine

5. Създайте празна директория c :\wbmachine\views\public\sites

6. Проверете дали имате файл

"C:\Program Files (x86)\GnuWin32\bin\wget.exe".

Ако нямаете, отворете

c :\wbmachine\scripts\process_pending_archives.php и заменете "C:\Program Files (x86)\GnuWin32\bin\wget.exe" с пътя до изпълнимия файл за командата wget.

7. Рестартирайте Apache web server

8. Свържете се към mysql базата от данни и изпълнете SQL заявките от файл c :\wbmachine\setup\db_create.sql

9. Използвайки Task Scheduler можете да пуснете периодично изпълнение на скрипта за изпълнение на заявките за архивиране
c:\wbmachine\scripts\process_pending_archives.php

Преди пускане на скрипта задайте следните environment variables:

```
$env:DBHOST = 'localhost'  
$env:DBNAME = 'wbmachine'  
$env:DBUSER = 'wbmachine'  
$env:DBPASS = 'ParolataESlozhna'  
$env:LOG_FILE = 'C:\wbmachine\wbmachine.log'
```

Пример за еднократно пускане:

```
> cd C:\wamp64\bin\php\php7.3.2  
> .\php.exe C:\wbmachine\scripts\process_pending_archives.php
```

10. Отворете localhost:3030. Ако видите началната страница на Wayback Machine, значи системата е успешно инсталриана

5.3. Конфигурация на системата

5.3.1. Конфигурация на сървърното приложение

Сървърното приложение поддържа следните опции за конфигурация, които се задават с environment variables:

- DBHOST - hostname, на който се намира базата от данни. Задължителна опция.
- DBNAME - име на базата от данни. Задължителна опция.
- DBUSER - име на потребител в базата от данни. Задължителна опция.
- DBPASS - парола на потребителя в базата от данни. Задължителна опция.
- S3_ENABLED - Приема стойности 1 (архивираните сайтове да се търсят на S3 bucket) или 0 (архивираните сайтове да се търсят на локалната файлова система)
- S3_URL - URL на S3 bucket, в който се архивират сайтовете (напр. <https://wbmachine.s3.amazonaws.com/>). Задължителна опция, ако S3_ENABLED=1

Ако инсталирате системата на Ubuntu 20.04, можете преди да пуснете инсталационния скрипт да промените параметрите във файл wbmachine/config/wbmachine.conf

5.3.2. Конфигурация на Process Pending Archives Timer

Скриптът за обработка на заявки за архивиране поддържа следните опции за конфигурация, които се задават с environment variables:

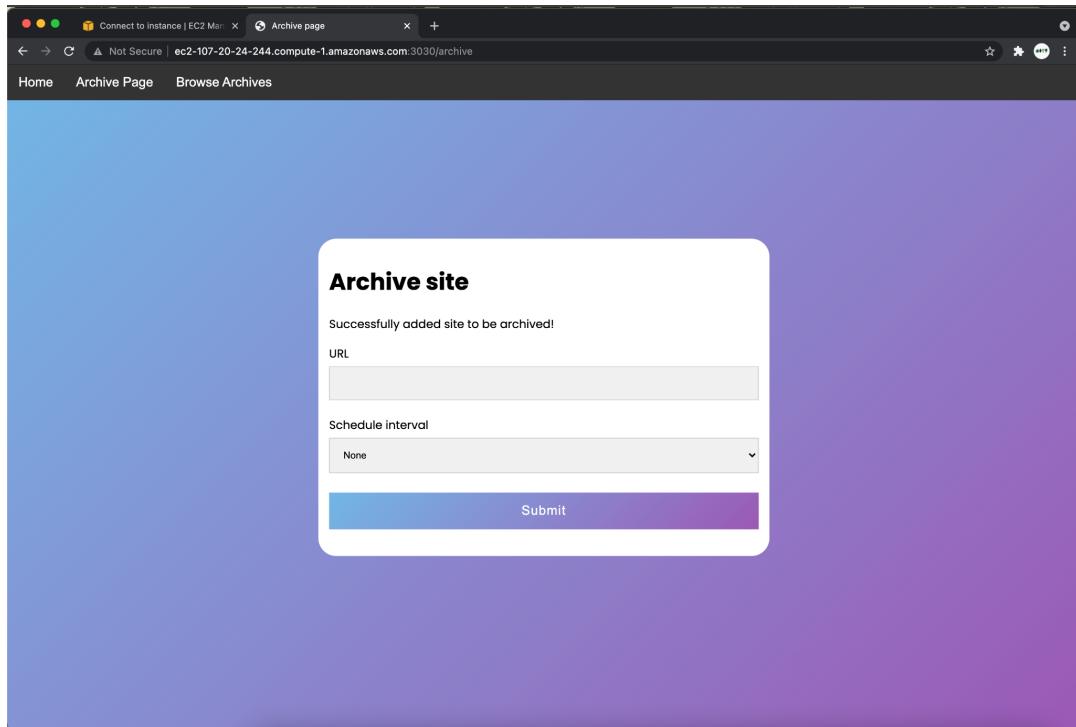
- DBHOST - hostname, на който се намира базата от данни. Задължителна опция.
- DBNAME - име на базата от данни. Задължителна опция.
- DBUSER - име на потребител в базата от данни. Задължителна опция.
- DBPASS - парола на потребителя в базата от данни. Задължителна опция.
- LOG_FILE - файл, в който се записва информация за свалянето на файлове, които са част от архивиран сайт. Задължителна опция.
- S3_ENABLED - Приема стойности 1 (сайтовете да се архивират в S3 bucket) или 0 (сайтовете да се архивират на локалната файлова система).
- S3_REGION - AWS регион, в който S3 bucket се намира. Задължителна опция, ако S3_ENABLED=1
- S3_LOCATION - S3 scheme URI на bucket (напр. "s3://wbmachine").
Задължителна опция, ако S3_ENABLED=1

Ако инсталирате системата на Ubuntu 20.04, можете преди да пуснете инсталационния скрипт да промените параметрите във файл wbmachine/config/wbmachine-process-pending-archives.service

6. Кратко ръководство на потребителя

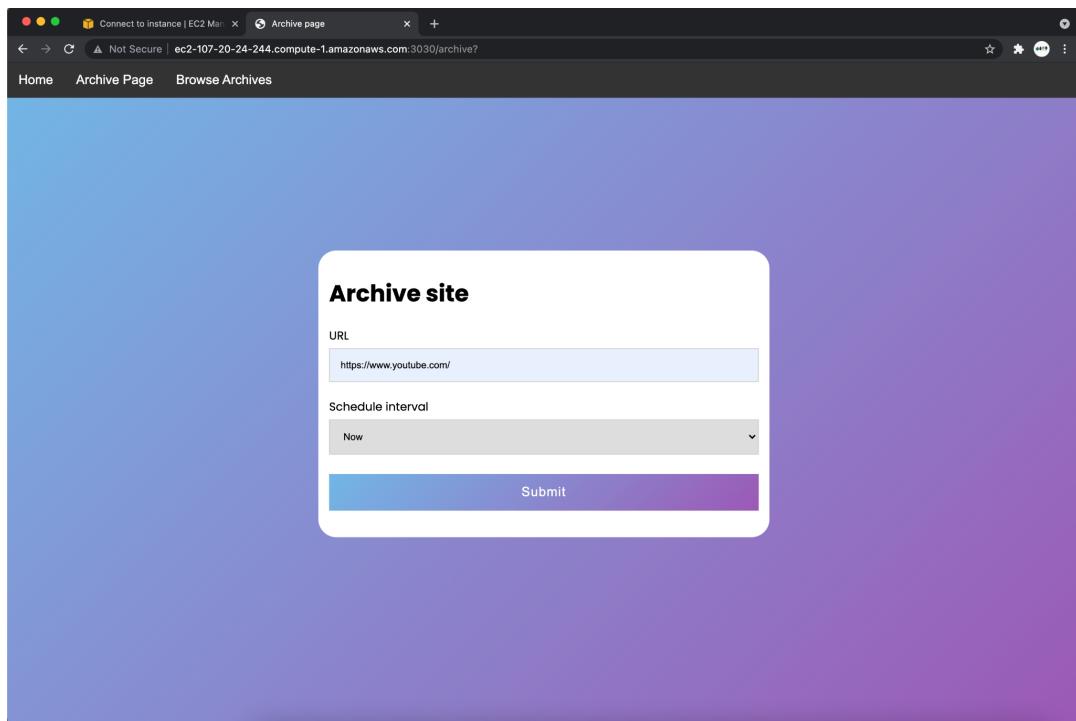
6.1. Архивиране на страница

1. Отворете страницата /archive:



фиг. 6 - Страницата за архивиране на сайт

2. В полето URL въведете адреса на страницата, която желаете да архивирате.
За Schedule Interval изберете Now.



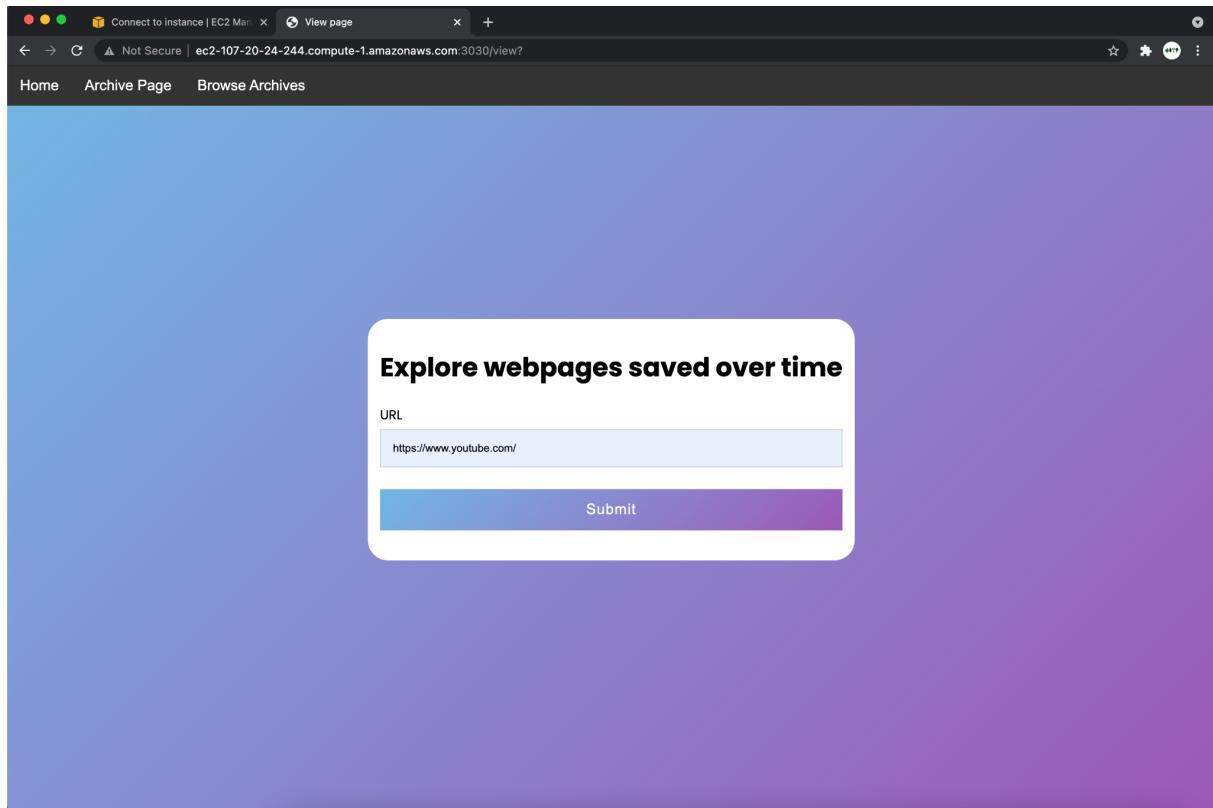
фиг. 7 - Страницата за архивиране с примерен вход

Натиснете бутона Submit.

Системата е планирала зададеното архивиране и след няколко минути ще можете да отворите архивираната страница.

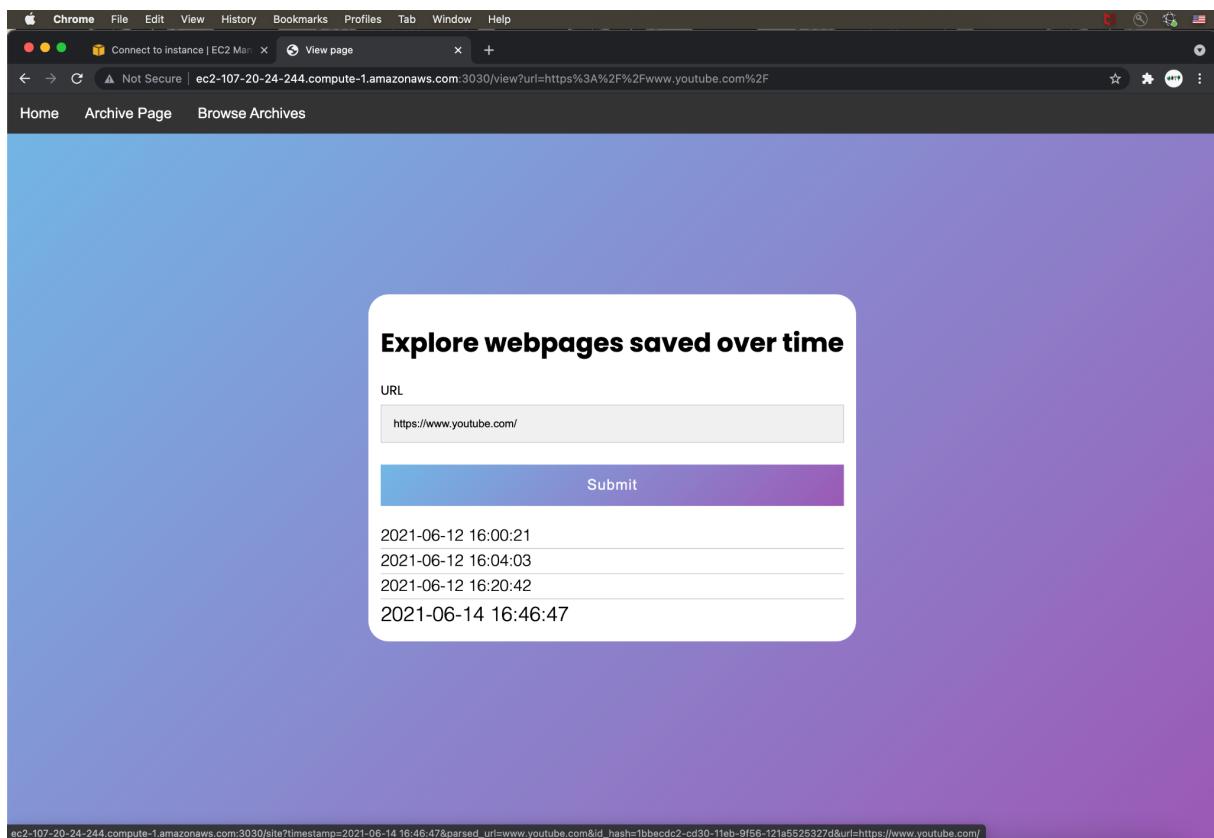
6.2. Отваряне на архивирана страница

1. Отворете страницата /view. В полето URL въведете адреса на архивираната страница, която желаете да отворите.

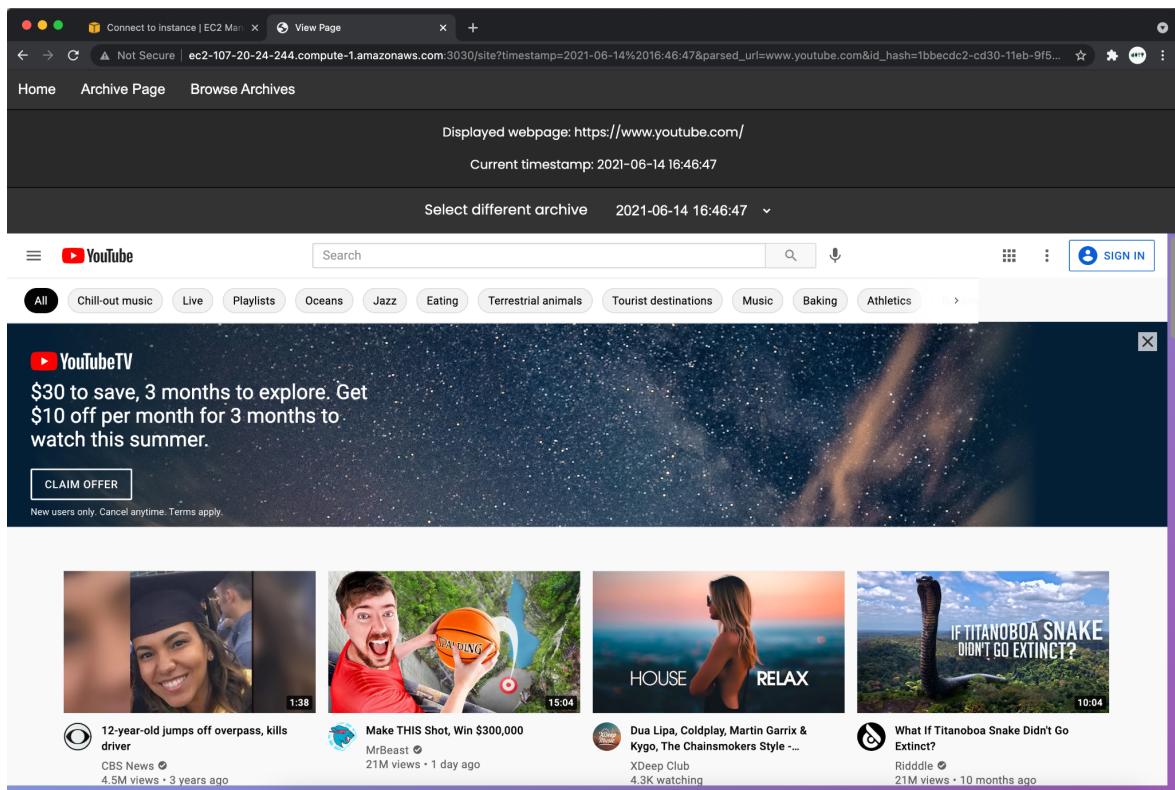


фиг. 8 - Страница за визуализация на сайт от архив

2. Ако в системата тази страница е добавена, ще ви се отвори списък с моменти на архивиране. Изберете момент от миналото, за да видите страницата в тогавашния ѝ вид.

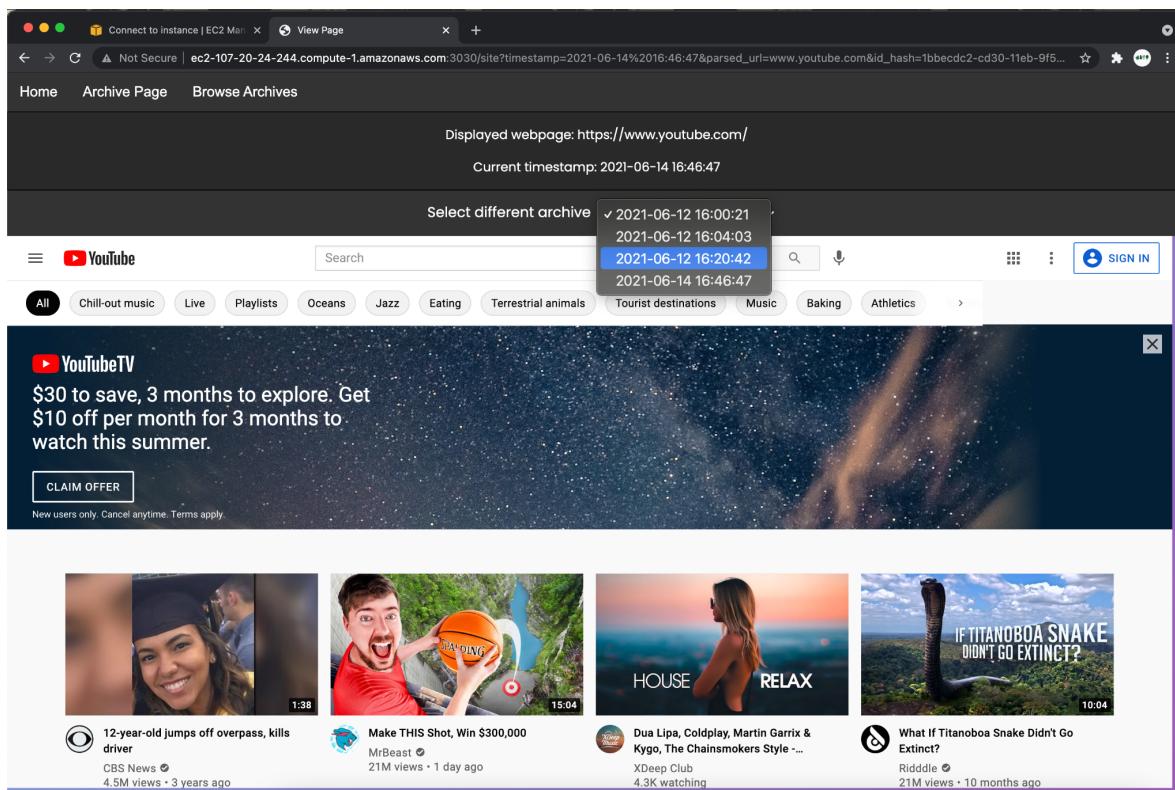


фиг. 9 - Страницата за визуализация на сайт с изброени възможни архиви за даден URL



фиг. 10 - Визуализация на примерна страница от архив

3. След отваряне на архивирана страница, можете да изберете друг момент на архивиране чрез опцията "Select different archive"



фиг. 11 - Избор на различен архив за визуализация

6.3. Използване на API на системата

6.3.1. Архивиране на сайтове

Заявки за архивиране на сайтове могат да се изпращат чрез заявка

POST /archive

Приемат се следните задължителни параметри:

- **url** - линк към страницата, която да се архивира
- **schedule_interval_id** - id на избрана настройката за архивиране

Поддържат се следните стойности за `schedule_interval_id`:

- 10 - прекрати периодичното архивиране
- 20 - архивирай сега
- 30 - архивирай веднъж месечно
- 40 - архивирай веднъж на 6 месеца
- 50 - архивирай веднъж годишно
- 60 - архивирай веднъж на 3 години
- 70 - архивирай веднъж на 5 години
- 80 - архивирай веднъж на 10 години

Отговор на заявката: статус код 200

6.3.2. Преглед на налични архиви за сайт

Заявки за преглед на данни за налични архиви за сайт могат да се изпращат чрез заявка

GET /view-api

Приема се следният параметър:

- **url** - линк на архивираната страница

Отговор на заявката: статус код 200 и JSON със следните полета:

- **status** - със стойност 'ok'
- **archives** - масив от обекти:
 - **timestamp** - момент на създаване на архив
 - **id_hash** - UUID на архив

6.3.3. Достъп до архивиран сайт

Заявки за достъп до архивиран сайт могат да се изпращат чрез заявка

GET /site-api

Приемат се следните параметри:

- **host** - URL на сървър, на който са разположени архивираните данни.
Незадължителен параметър. По подразбиране се взима сървъра, който е предоставил страницата, от която се прави заявката.
- **url** - линк на архивираната страница.
- **id_hash** - UUID на архив

Задължително трябва да бъде зададен или url, или id_hash. Ако и двете са зададени, url се игнорира.

Отговор на заявката: статус код 302 и header "Location" за пренасочване към архивираната страница

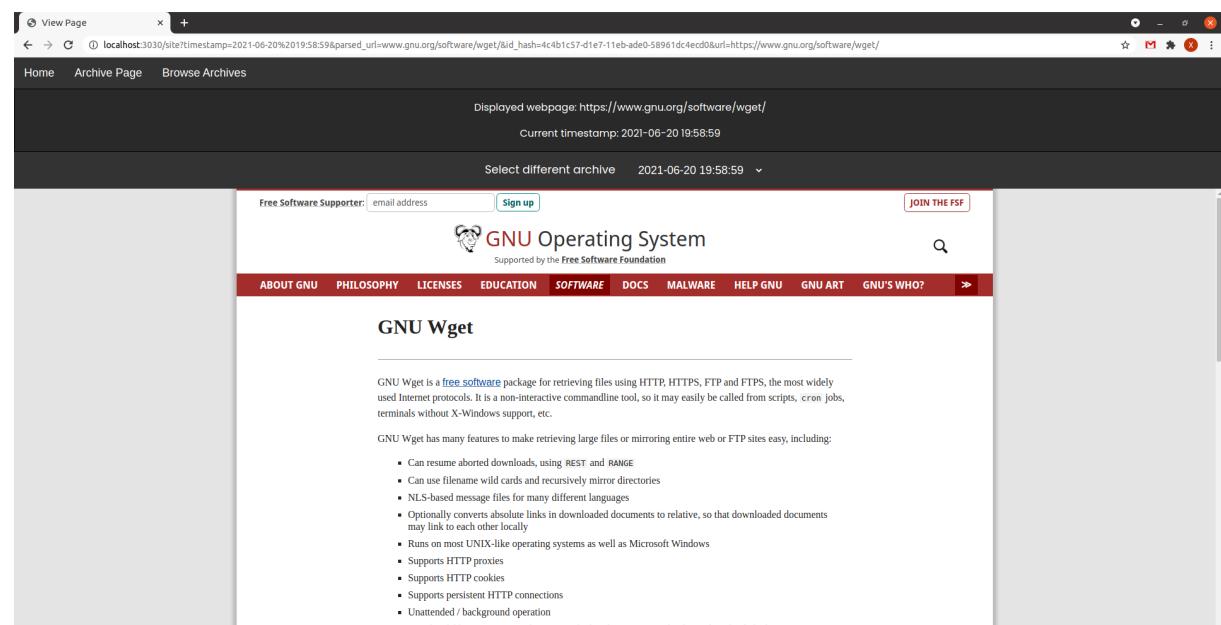
7. Примерни данни

За демонстрация на работата на системата с примерни данни използвахме следната страница:

<https://www.gnu.org/software/wget/>

След инсталация на системата можете като суперпотребител да изпълните скрипта `./setup/db_fill.sh` или ръчно да изпълните в базата заявките в `./setup/db_fill.sql`

Отворете страницата Browse Archives, сложете в полето URL „<https://www.gnu.org/software/wget/>“, натиснете Submit, изберете timestamp и архивираната страница ще се визуализира



фиг. 12 - Пример за визуализация на архивиран сайт

8. Описание на програмния код

8.1. Описание на структурата на проекта

- **doc** - диаграми и друга документация
- **setup** - директория с инсталационни скриптове
 - **db_create.sql** - създава базата
 - **db_drop.sql** - трне базата
 - **db_fill.sql** - попълва базата с данни
 - **install.sh** - прави setup на системата
 - **uninstall.sh** - деинсталира системата
- **server** - директория с бизнес логика
 - **modules** - директория с PHP модули
 - **index.php** - входната точка за сървъра, в която са дефинирани route-овете
- **views** - директория с логика за визуализация на клиентското приложение
 - **pages** - изгледи за генериране на страници
 - **public** - директория, която е публична за клиента
 - **js** - js файлове
 - **css** - css файлове
 - **layout.html** - главният изглед
- **config** - конфигурационни файлове
- **scripts** - директория със скриптове

9. Приноси на студента, ограничения и възможности за бъдещо разширение

9.1. Приноси на студента

Христо Бориславов Спасов

- Проектиране и изграждане на архитектурата на системата
- Проектиране и изграждане на базата от данни
- Проектиране и изграждане на логиката за архивиране на сайтове
- Създаване на инсталационни скриптове
- Създаване на API за изпращане на заявки за архивиране и достъп до архивирани страници

Таня Начева Желева

- Проектиране на layout на системата
- Изграждане на фронт-енд на системата
- Проектиране и изграждане на логиката за разглеждане на архиви
- Проектиране и изграждане на функционалността за визуализация на архиви

9.2. Ограничения

- Настоящата версия позволява архивиране на страници само поединично и не се поддържа механизъм за свободно сърфиране между тях. Дори и

да се изгради подобен механизъм, той задължително трябва да включва ограничение за количеството свалени ресурси.

- Системата не поддържа архивиране на страници, които съдържат query параметри.
- Системата не поддържа надеждно архивиране на страници, които се генерират чрез javascript чрез изпращане на заявки до други сайтове. Подобна функционалност може да се реализира чрез изпълнение на javascript, който взема зареденото на страницата съдържание.

9.3. Възможности за бъдещо развитие

- Възможност за архивиране на цели сайтове или част от тях със зададено ограничение
- Възможност за скрити архивирани страници, които се достъпват с парола
- Асинхронно изпълнение на скрипта за обработка на заявки за архивиране с цел оптимизация
- Поддръжка на няколко worker-а, които да могат да обработват заявки за архивиране на страници
- Добавяне на вътрешен механизъм за сравняване на всеки нов архив със стар такъв, за да се намали количеството пазени данни
- Възможност за избор в коя директория да се съхраняват архивираните сайтове
- Добавяне на ограничение за количество ресурси, които се свалят при архивиране на сайт
- Показване на thumbnail на сайтове
- Опция дали да се свалят всички ресурси или само HTML
- Да не се създават всички директории за ресурсите в една директория, а да има разпределение по първите две букви от идентификатора

10. Какво научих

- Git Feature Branch Workflow
- php ini
- Дефиниране на константи в PHP
- require, include и use в PHP
- PHP функциите print_r и error_log, чрез които удобно може да се прави TRACE
- PHP функцията parse_url
- PHP regex
- PHP функцията за проверка дали променлива е зададена: empty.
Разликата между empty, isset
- Как се взимат параметри при POST заявка: от \$_POST
- Как се взимат параметри при GET заявка: от \$_GET
- \$_SERVER - информация за заявката и за сървъра - хедъри, протокол, HTTP метод, порт и други
- Apache VirtualHost конфигурация
- PHP алтернативен синтаксис за контролни структури

- Работа с wget - опциите за сваляне на всички ресурси от страницата и конвертиране на линковете за тях да са линкове към локално свалените ресурси
- Използване на WAMP
- shell_exec в PHP под Windows 10 използва cmd

11. Използвани източници

[1] **Wget - The non-interactive network downloader.**

[<https://linux.die.net/man/1/wget>]

[2] **systemd.timer — Timer unit configuration**

[<https://www.freedesktop.org/software/systemd/man/systemd.timer.html>]

[3] **PHP Manual** [<https://www.php.net/manual/en/>]

[4] **VirtualHost Examples, Apache HTTP Server Version 2.4**

[<https://httpd.apache.org/docs/2.4/vhosts/examples.html>]

Предал (подпись):

/62278, Христо Спасов, СИ, З/

Предал (подпись):

/62288, Таня Желева, СИ, З/

Приел (подпись):

/доц. Милен Петров/