A Project Report on

BE TECHY

Submitted by

Salonee Powar(19IT5003)

Shweta Kadam(19IT5005)

Sonali Malage(19IT5013)

Under the guidance of

Mrs. Anita Senathi

Assistant Professor
Department of Information Technology
Ramrao Adik Institute of Technology
Nerul, Navi Mumbai

Certificate

This is to certify that the project entitled **Be Techy (News Aggregator)** is being submitted to the Department of Information Technology, Ramrao Adik Institute of Technology, Navi Mumbai.

Project Guide   External
Examiner (Mrs. Anita Senathi)   (   )

Acknowledgement

We owe our gratitude to many people who have supported us throughout this journey.

We would, firstly like to express our heartfelt gratitude towards our respected Principal Dr. Mukesh Patil and our Head of Department Dr. Ashish Jadhav for providing us immense facilities, guidance and never ending support.

The completion of any inter-disciplinary project depends upon cooperation and combined efforts of several sources of knowledge. We take this opportunity to express our profound gratitude and deep regards to our guide Mrs. Anita Senathi for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

Lastly, we thank our parents, family, friends and well wishers who always looked for the chance to help us in whatever means came forth and for their constant encouragement without which the project would not be a distant reality.

Introduction

News aggregator service is a very important start of the day. There are various publications and news sites online. They publish their content on multiple platforms. Now, imagine when you open 10-20 news sites every day. The time you waste to gain information. Information gain is everything in today's world.

In proposed system more and more people value the easy access and availability of online content on gadgets and technology. This is because such information provides great value and, most of all, it's free.

This project's main idea is to develop a web application which aggregates data (Technology related news articles) from multiple websites. Then presents the data in one location.

Proposed System

More and more people value the easy access and availability of online content on gadgets and technology. This is because such information provides great value and, most of all, it's free. One thing that works in favour of online content is that news and information, including technology and gadget articles, are fresh and evergreen. By the time that the content hits print media such as journals and magazines, the information has become stale.

In proposed system more and more people value the easy access and availability of online content on gadgets and technology. This is because such information provides great value and, most of all, it's free.

The main objective of the proposed system is to provide easy access of technology and gadgets news articles via website.

So, our news aggregator will work in 3 steps:

- ◦ It scrapes the web for news articles.
- ◦ Then it stores the articles.
- ◦ The stored articles are served to the client in a nice template.

System Components

1. **Frontend:** HTML5, CSS, Bootstrap
2. **Backend**: Python, Django framework .0.4, News API: Python client library
3. **Database**: Dbsqlite3

1.1 HTML:

HTML is HyperText Mark-up Language used to documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. User interface is designed using HTML CSS and Bootstrap.

1.2 CSS:

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

1.3 Bootstrap:

Bootstrap is a free front-end framework for faster and easier web development
Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins

2.1 Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. The fast edit-test-debug cycle makes this simple approach very effective.

2.2 Django (3.0.4):

Django is a free and open-source web framework, written in Python, which follows the model-view-template architectural pattern. In our project we have used this framework to host our leave management system in which users will login to apply for leave that will be approved or not approved by admin.

Django is a widely-used Python web application framework with a "batteries-included" philosophy. The principle behind batteries-included is that the common functionality for building web applications should come with the framework instead of as separate libraries. Authentication, URL routing, a template engine, an object-relational mapper (ORM), and database schema migrations are all included with

the Django framework. Compare that included functionality to the Flask framework which requires a separate library such as Flask-Login to perform user authentication.

Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.

2.3 Newsapi:

News API is a simple HTTP REST API for searching and retrieving live articles from all over the web. News API is a simple and easy-to-use API that returns JSON metadata for headlines and articles live all over the web right now.

Authentication is handled with a simple API key. Python client library to integrate News API into your Python application
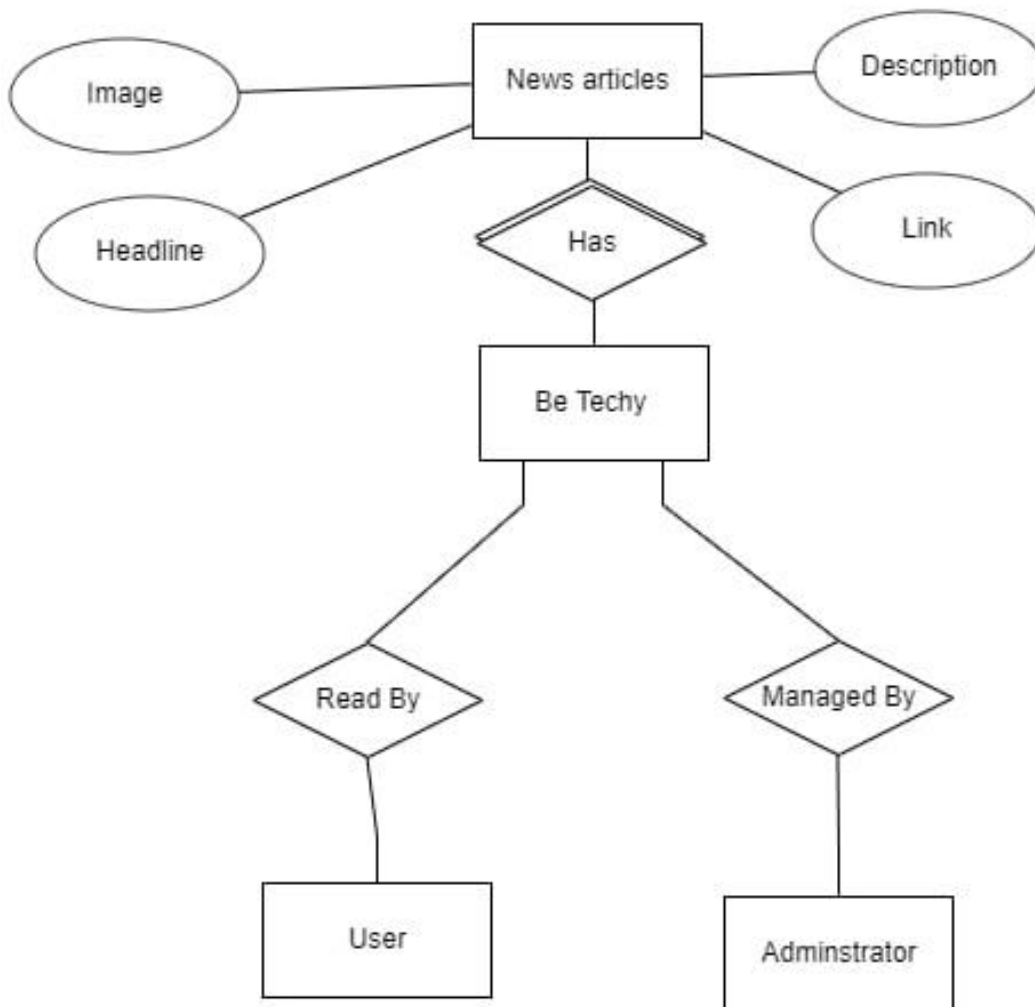
$ pip install newsapi-python

3.1 Dbsqlite3:

SQLite is an in-process library that implements a self-contained, server less, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects. Dbsqlite3 is a built-in database provided Django framework.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.

Architecture

Be Techy (ER Diagram)

WORKING OF SYSTEM

The **Be Techy** basically consist of two modules:

1. Home page

It contains different Technology related news articles which are aggregated from multiple websites.

2. Different six websites

These websites are:

- Techradar: TechRadar is an online publication focused on technology, with editorial teams in the US, UK, Australia and India. It provides news and reviews of tech products.

- Techcrunch: TechCrunch is an American online publisher focusing on the tech industry. The company specifically reports on the business related to tech, technology news.

- Wired: Wired is a monthly American magazine, published in print and online editions, that focuses on how emerging technologies affect culture, the economy, and politics.

- The Next Web: The Next Web (TNW) is a website focused on news from around the world. Internet of things, development, coding, infrastructure, security, cloud.

- The Verge: The Verge is an American technology-news online magazine operated by Vox Media, publishing news, feature stories, guidebooks, product reviews and so.

- Engadget: Engadget is a multilingual technology blog network with daily coverage of gadgets and consumer electronics.

Each website has different News Articles.

Advantages of Be Techy

1. The information provides great value and, most of all, it's free.

2. The speed of delivery of quality online technology content is fast through the Internet.

3. The easy access of technology and gadgets content via website.

Django Source Code

1. NewsApp\views.py:

```python
from django.shortcuts import render
from newsapi import NewsApiClient

def index(request):
    newsapi = NewsApiClient(api_key="b0f75ce660c0466a9a98c2478f8abb62")
    topheadlines =
newsapi.get_top_headlines(sources='TechRadar,techcrunch,wired,engadget,the-next-
web,the-verge')

    articles = topheadlines['articles']
    desc = []
    news = []
    img = []
    ur = []

    for i in range(len(articles)):
        myarticles = articles[i]

        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        ur.append(myarticles['url'])

    mylist = zip(news, desc, img, ur)
    return render(request, 'index.html', context={"mylist":mylist})

def TechRadar(request):
    newsapi = NewsApiClient(api_key="b0f75ce660c0466a9a98c2478f8abb62")
    topheadlines = newsapi.get_top_headlines(sources='TechRadar')

    articles = topheadlines['articles']
    desc = []
    news = []
```

```python
    img = []
    ur = []

    for i in range(len(articles)):
        myarticles = articles[i]
        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        ur.append(myarticles['url'])
    mylist = zip(news, desc, img, ur)
    return render(request, 'TechRadar.html', context={"mylist":mylist})

def techcrunch(request):
    newsapi = NewsApiClient(api_key="b0f75ce660c0466a9a98c2478f8abb62")
    topheadlines = newsapi.get_top_headlines(sources='techcrunch')

    articles = topheadlines['articles']
    desc = []
    news = []
    img = []
    ur = []

    for i in range(len(articles)):
        myarticles = articles[i]
        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        ur.append(myarticles['url'])

    mylist = zip(news, desc, img, ur)

    return render(request, 'techcrunch.html', context={"mylist":mylist})
#google-news,engadget,business-insider,wired,the-times-of-india,techcrunch,the-verge,the-next-web

def wired(request):
    newsapi = NewsApiClient(api_key="b0f75ce660c0466a9a98c2478f8abb62")
    topheadlines = newsapi.get_top_headlines(sources='wired')
```

```python
    articles = topheadlines['articles']
    desc = []
    news = []
    img = []
    ur = []

    for i in range(len(articles)):
        myarticles = articles[i]

        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        ur.append(myarticles['url'])

    mylist = zip(news, desc, img, ur)
    return render(request, 'wired.html', context={"mylist":mylist})

def theverge(request):
    newsapi = NewsApiClient(api_key="b0f75ce660c0466a9a98c2478f8abb62")
    topheadlines = newsapi.get_top_headlines(sources='the-verge')

    articles = topheadlines['articles']
    desc = []
    news = []
    img = []
    ur = []

    for i in range(len(articles)):
        myarticles = articles[i]

        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        ur.append(myarticles['url'])

    mylist = zip(news, desc, img, ur)
    return render(request, 'theverge.html', context={"mylist":mylist})

def thenextweb(request):
```

```python
    newsapi = NewsApiClient(api_key="b0f75ce660c0466a9a98c2478f8abb62")
    topheadlines = newsapi.get_top_headlines(sources='the-next-web')

    articles = topheadlines['articles']
    desc = []
    news = []
    img = []
    ur = []

    for i in range(len(articles)):
        myarticles = articles[i]

        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        ur.append(myarticles['url'])

    mylist = zip(news, desc, img, ur)
    return render(request, 'thenextweb.html', context={"mylist":mylist})

def engadget(request):
    newsapi = NewsApiClient(api_key="b0f75ce660c0466a9a98c2478f8abb62")
    topheadlines = newsapi.get_top_headlines(sources='engadget')

    articles = topheadlines['articles']
    desc = []
    news = []
    img = []
    ur = []
    for i in range(len(articles)):
        myarticles = articles[i]

        news.append(myarticles['title'])
        desc.append(myarticles['description'])
        img.append(myarticles['urlToImage'])
        ur.append(myarticles['url'])

    mylist = zip(news, desc, img, ur)
    return render(request, 'engadget.html', context={"mylist":mylist}
```

2. NewsApp\urls.py:

```python
from django.urls import path
from .views import index,TechRadar, techcrunch, wired ,theverge, thenextweb, engadget

urlpatterns = [
path('', index, name = 'index'),
path('TechRadar/', TechRadar, name = 'TechRadar'),
path('techcrunch/', techcrunch, name = 'techcrunch'),
path('wired/', wired, name = 'wired'),
path('theverge/', theverge, name = 'theverge'),
path('thenextweb/', thenextweb, name = 'thenextweb'),
path('engadget/', engadget, name = 'engadget'),


]
```

3. News\urls.py:

```python
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('NewsApp.urls'))
]
```

4. Homepage:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Be Techy</title>
    <style>body.solid {border-style: solid;}</style>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
 <style type="text/css"> @media (min-width: 1200px) {
   .container{
       max-width: 1290px;
   }
}
span { display: flex;
     justify-content: center }


.text {
   font-family: verdana;
   font-size: 6em;
   font-weight: 700;
   color: white;
   text-shadow: 1px 1px 1px #919191,
      1px 2px 1px #919191,
      1px 3px 1px #919191,
      1px 4px 1px #919191,
      1px 5px 1px #919191,
      1px 6px 1px #919191,
      1px 7px 1px #919191,
      1px 8px 1px #919191,
      1px 9px 1px #919191,
      1px 10px 1px #919191,
     1px 18px 6px rgba(16,16,16,0.4),
     1px 22px 10px rgba(16,16,16,0.2),
     1px 25px 35px rgba(16,16,16,0.2),
     1px 30px 60px rgba(16,16,16,0.4);
}
</style>
</head>
```

```html
<body class="solid" >
{% load static %}
<div class="jumbotron" style="background-image: url({% static 'tc.jpg' %});
color:black,min-height: 300;">

{% load static %}
<img align="left" src="{% static "bte.png" %}" alt="BeTechy" style="margin-top:-25px";
height="135" width="155">

  <span class="text">
    BE TECHY: Latest News
  </span>
</div>

<center>
  <a class="btn btn-danger btn-lg" role="button" href="http://127.0.0.1:8000/TechRadar"
style=" font-weight: 500;font-size: 28px; border-radius: 10px; border: none;  box-shadow:
0px 8px 15px rgba(0, 0, 0, 0.2); ">Techradar</a>     
  <a class="btn btn-primary btn-lg" role="button" href="http://127.0.0.1:8000/techcrunch"
style="font-weight: 500;font-size: 28px; border-radius: 10px; border: none;  box-shadow:
0px 8px 15px rgba(0, 0, 0, 0.2); ">Techcrunch</a>    
  <a class="btn btn-danger btn-lg" role="button" href="http://127.0.0.1:8000/wired"
style="font-weight: 500;font-size: 28px; border-radius: 10px; border: none;  box-shadow:
0px 8px 15px rgba(0, 0, 0, 0.2);">Wired</a>     
<a class="btn btn-primary btn-lg" role="button" href="http://127.0.0.1:8000/theverge"
style="font-weight: 500;font-size: 28px; border-radius: 10px; border: none;  box-shadow:
0px 8px 15px rgba(0, 0, 0, 0.2);">The Verge</a>    
<a class="btn btn-danger btn-lg" role="button" href="http://127.0.0.1:8000/thenextweb"
style="font-weight: 500;font-size: 28px; border-radius: 10px; border: none;  box-shadow:
0px 8px 15px rgba(0, 0, 0, 0.2);">The Next Web</a>    
<a class="btn btn-primary btn-lg" role="button" href="http://127.0.0.1:8000/engadget"
style="font-weight: 500;font-size: 28px; border-radius: 10px; border: none;  box-shadow:
0px 8px 15px rgba(0, 0, 0, 0.2);">Engadget</a>
<br>
</center><br><br>
<div class="container">
  <div class="row" >

{% for new, des,i,ur in mylist %}
```

```html
<div class="col-lg-4" >
<div class="thumbnail" style="border: none;  box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.9);">

<a href="{{ur}}" target="_blank">
<br><h1> {{new}}</h1> <br>
<img src="{{i}}" alt="" width="390" height="240">
</a>
<br><div class="caption" style="font-size: 20px;">
<b>Description: </b>{{des}}
</div></div></div>
{% endfor %}
</div></div>
</body>
</html>
```
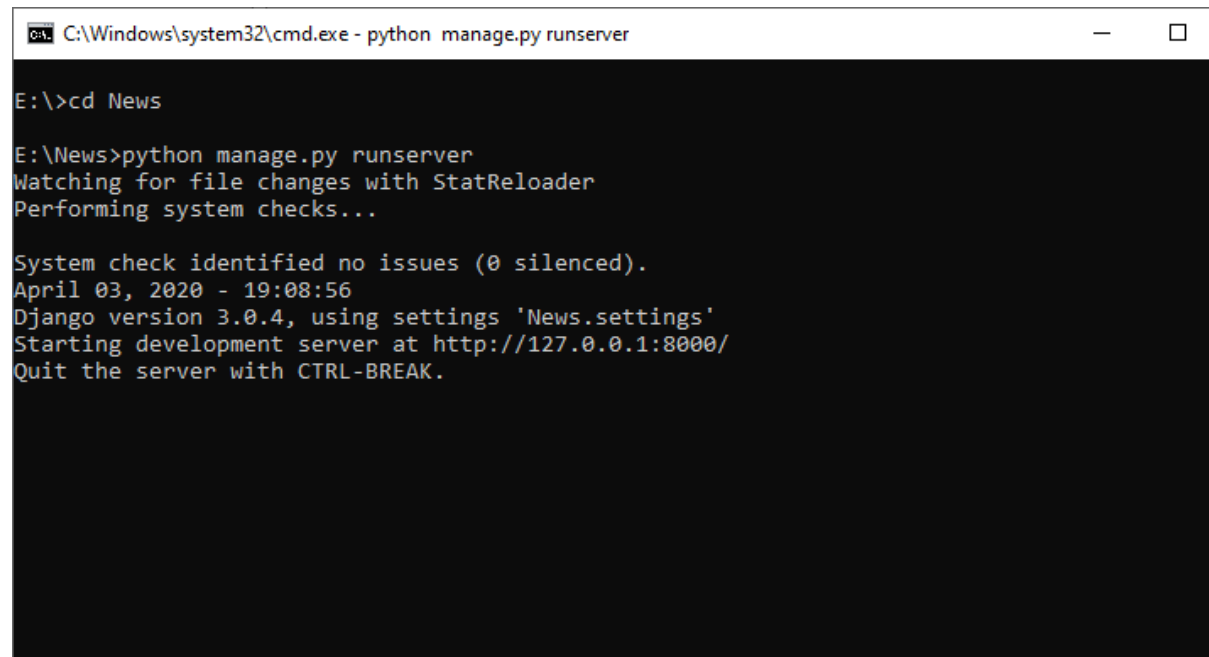
5. Techcrunch:

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Techcrunch News</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
 <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
 <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
  <style type="text/css"> @media (min-width: 1200px) {
  .container{
    max-width: 1290px;
```

```css
        }
    }
span { display: flex;
        justify-content: center }

.text {

    font-family: verdana;
    font-size: 6em;
    font-weight: 700;
    color: white;
    text-shadow: 1px 1px 1px #919191,
        1px 2px 1px #919191,
        1px 3px 1px #919191,
        1px 4px 1px #919191,
        1px 5px 1px #919191,
        1px 6px 1px #919191,
        1px 7px 1px #919191,
        1px 8px 1px #919191,
        1px 9px 1px #919191,
        1px 10px 1px #919191,
      1px 18px 6px rgba(16,16,16,0.4),
      1px 22px 10px rgba(16,16,16,0.2),
      1px 25px 35px rgba(16,16,16,0.2),
      1px 30px 60px rgba(16,16,16,0.4);
}
</style>
</head>
<body class="solid" >
{% load static %}
<div class="jumbotron" style="background-image: url({% static 'tc.jpg' %});
color:black,min-height: 300;">

{% load static %}
<img align="left" src="{% static "techcrunch.png" %}" alt="Techcrunch" style="margin-
top:-0px"; height="80" width="350">

    <span class="text">
        Techcrunch News
```

```html
    </span>
</div>

<div class="container">
  <div class="row">

{% for new, des,i,ur in mylist %}
<div class="col-lg-4" >
<div class="thumbnail" style="border: none;  box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.9);">

<a href="{{ur}}" target="_blank">
<br><h1> {{new}}</h1> <br>
<img src="{{i}}" alt="" width="390" height="240">
</a>
<br><div class="caption" style="font-size: 20px;">
<b>Description: </b>{{des}}
   </div></div></div>
{% endfor %}
</div></div>
</body>
</html>
```

Same for other 5 websites.

Implementation of Web Application

1. Start the Django Web Server.

2. To check the server is running or not type localhost:8000 in browser (using Google chrome here). If the server is running the Home page of website will be displayed.

3. After clicking on Techcrunch:

TechCrunch

Techcrunch News

T-Mobile customers on unlimited wireless family plans get a free year of Quibi

Description: T-Mobile this morning officially announced its exclusive partnership with the new streaming service, Quibi, set to launch on April 6. The service will be made available for free for a year to T-Mobile customers on its unlimited wireless family plans. The stre...

Forward launches 'Forward At Home' primary care service to address COVID-19 healthcare crunch

Description: The global coronavirus pandemic has already caused a tremendous strain on healthcare resources around the world, and it's leading to a shift in how healthcare is offered. Startup Forward, which debuted in 2016 and has since expanded its tech-focused primary c...

Venture-backed Celularity receives FDA approval for early trials of a new cell therapy for COVID-19

Description: Celularity, the venture-backed developer of novel cell therapies for cancer treatments, has received an initial clearance from the Food and Drug Administration to begin early-stage clinical trials on a potential treatment for COVID-19. The company, which has ...

Amazon begins running temperature checks and will provide surgical masks at warehouses

Description: Amazon has detailed someone measures its taking to prevent any further spread of COVID-19 at its warehouse facilities in the U.S. and Europe, according to Reuters, including taking temperature checks and distributing facemasks to employees at Amazon warehous...

Jim Fruchterman raises $1.7M for Tech Matters, a new effort to help nonprofits do tech better

Description: Social entrepreneurship pioneer Jim Fruchterman has launched a new nonprofit, Tech Matters, with $1.7 million in backing from corporate and foundation sources, including Twilio, Okta, Working Capital, Facebook and Schmidt Futures. Tech Matters is Fruchterman'...

A bug bounty alone won't save your startup -- here's why

Description: For startups and companies, these are the dos and don'ts of vulnerability disclosure.

Conclusion

It is a web application which aggregates data (Technology related news articles) from multiple websites. Then presents the data in one location. you are just a click away to get information from various websites. The information provides great value and, most of all, it's free.

References

- Django documentation (3.0.4)
- The Definitive Guide to Django Web Development Done Right by Adrian Kaplan-Moss, Jacob
- https://codeloop.org/
- https://spoken-tutorial.org
- https://simpleisbetterthancomplex.com
- Sentdex Youtube