JARVIS:THE NEWS APP

A MINI PROJECT REPORT

Submitted by

JITHEESWARAN.V	**220701108**
JEROME SATHIYAGU.A	**220701106**
JAYA PRAKASH.V	**220701103**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

2024 - 25

BONAFIDE CERTIFICATE

Certified that this project report “**JARVIS:THE NEWSAPP**” isthe bonafide work of **“JITHEESWARAN V (220701108) ,JEROME SANTHIYAGU.A(220701106),JAYA PRAKASH.V(220701103) ”** who carried out the project work under my supervision.

Submitted for the Practical Examination held on ________________________

SIGNATURE	**SIGNATURE**
Dr.R.SABITHA	**Ms. DHARANI DEVI M.Tech.,PhD.,**
Professor and II Year Academic Head	**Associate Professor (SG),**
ComputerScience and Engineering,	**Computer Science and Engineering,**
Rajalakshmi Engineering College	**Rajalakshmi Engineering College,**
(Autonomous),	**(Autonomous),**
Thandalam, Chennai - 602 105	**Thandalam, Chennai - 602 105**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

JarvisNewsApp is a cutting-edge news aggregation application designed to keep users informed and connected with global events. Offering a seamless interface and personalized news feeds, JarvisNewsApp curates content from diverse sources, ensuring comprehensive coverage across various topics and regions. With features like real-time updates, customizable filters, and intuitive navigation, users can effortlessly access relevant news tailored to their interests. JarvisNewsApp redefines the news consumption experience, empowering users to stay abreast of current affairs and engage with the world around them like never before.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR.S.MEGANATHAN** and the chairperson **DR.M.THANGAM MEGANATHAN** for their timely support and encouragement. We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance. No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr.P.KUMAR M.E Ph.D.,** for being ever supporting force during our project work We also extend our sincere and hearty thanks to our internal guide Dr G **DHARANI DEVI M.Tech.,PhD.,** for her valuable guidance and motivation during the completion of this project. Our sincere thanks to our **JARVIS INDUSTRIES** team and other staff members of computer science engineering who worked with us in this mini project.

1.JITHEESWARAN.V

2.JEROME SATHIYAGU.A

3.JAYAPRAKESH.V

TABLE OF CONTENTS

1. INTRODUCTION

    1.1 INTRODUCTION

    1.2 OBJECTIVES

    1.3 MODULES

2. SURVEY OF TECHNOLOGIES

    2.1 SOFTWARE DESCRIPTION

    2.2 LANGUAGES

        2.2.1 MONGODB

        2.2.2 HTML,CSS,JS

        2.2.3 PYTHON

3. REQUIREMENTS AND ANALYSIS

    3.1 REQUIREMENT SPECIFICATION

    3.2 HARDWARE AND SOFTWARE REQUIREMENTS

    3.3 ARCHITECTURE DIAGRAM

    3.4 ER DIAGRAM

    3.5 NORMALIZATION

4. PROGRAM CODE

5. RESULTS AND DISCUSSION

6. CONCLUSION

7. REFERENCE

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

In an era characterized by information overload and rapid technological advancements, staying informed about global events has become more crucial than ever. JarvisNewsApp emerges as a solution to address this need, offering users a streamlined platform to access curated news content from diverse sources.With the proliferation of digital media, traditional methods of news consumption have evolved. JarvisNewsApp recognizes the importance of providing users with personalized news feeds tailored to their preferences and interests. By leveraging sophisticated algorithms, JarvisNewsApp ensures that users receive relevant and timely updates on topics ranging from politics and technology to sports and entertainment.The user-centric design of JarvisNewsApp prioritizes accessibility and ease of use. Real-time updates and customizable filters empower users to navigate through vast amounts of information efficiently, allowing them to stay abreast of the latest developments without feeling overwhelmed.Through this introduction, we aim to provide an overview of JarvisNewsApp's mission and functionality. Subsequent sections will delve deeper into the features, technology stack, and future enhancements of the application, showcasing its potential to revolutionize the way users engage with news content in today's fast-paced digital landscape.

1.2 OBJECTIVE

Our primary objective with JarvisNewsApp is to offer users a seamless and personalized news consumption experience. Through an intuitive interface, users can easily access curated news content tailored to their interests and preferences. We aim to ensure comprehensive coverage of diverse topics and regions, enabling users to stay informed about global events. Additionally, JarvisNewsApp strives to enhance user engagement by providing real-time updates and customizable features, thereby fostering a deeper connection with the news and the world around them.

1.3 MODULE

- DAILY NEWS APP
- CATERIGORISHED COUNTRY NEWS
- LANGAUGE CHANGED
- DATAS SAVES IN MONGODB
- IF USER WANTS TO SEE EARLY NEWS THEY CAN SEARCH THE NEWS BY DATES
- LINK IS ALSO SAVED LIVE

Ch 2. SURVEY OF TECHNOLOGY

2.1 SOFTWARE DESCRIPTION

Visual studio Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

2.2 LANGUAGES

2.2.1 HTML

Hypertext: text (often with embeds such as images, too) that is organized in order to connect related items

Markup: a style guide for typesetting anything to be printed in hardcopy or soft copy format

Language: a language that a computer system understands and uses to interpret commands.

HTML determines the structure of web pages. This structure alone is not enough to make a web page look good and interactive. So you'll use assisted technologies such as CSS and JavaScript to make your HTML beautiful and add interactivity, respectively.

2.2.2 CSS

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.

2.2.3 PYTHON

Python, a high-level programming language, serves as the backbone of JarvisNewsApp's backend functionality. Leveraging its simplicity and versatility, Python enables rapid development and deployment of web applications. Within JarvisNewsApp, Python processes user requests, interacts with the MongoDB database to retrieve and manipulate news articles, and dynamically generates HTML content for presentation on the frontend. Its extensive library ecosystem provides access to tools and frameworks essential for building robust and scalable web applications, ensuring JarvisNewsApp operates seamlessly.

2.2.4 MONGODB

MongoDB operates as a document-oriented NoSQL database, storing data in flexible JSON-like documents. Each document can have its own unique structure, allowing for dynamic and scalable data modeling. MongoDB employs indexes to facilitate fast querying and retrieval of data, optimizing performance even with large datasets. Its distributed architecture enables horizontal scaling, ensuring seamless handling of increasing data volumes. Additionally, MongoDB offers features like sharding and replication for high availability and fault tolerance in distributed environments.

Ch 3. REQUIREMENT AND ANALYSIS

3.1 REQUIREMENTS SPECIFICATION

User Requirements

The system requirement in library management focuses on the possibility of search for books by title, author or subject by the member.

System Requirements

There should be a database backup of the library management system. Operating system should be WindowsXP or a higher version of windows.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

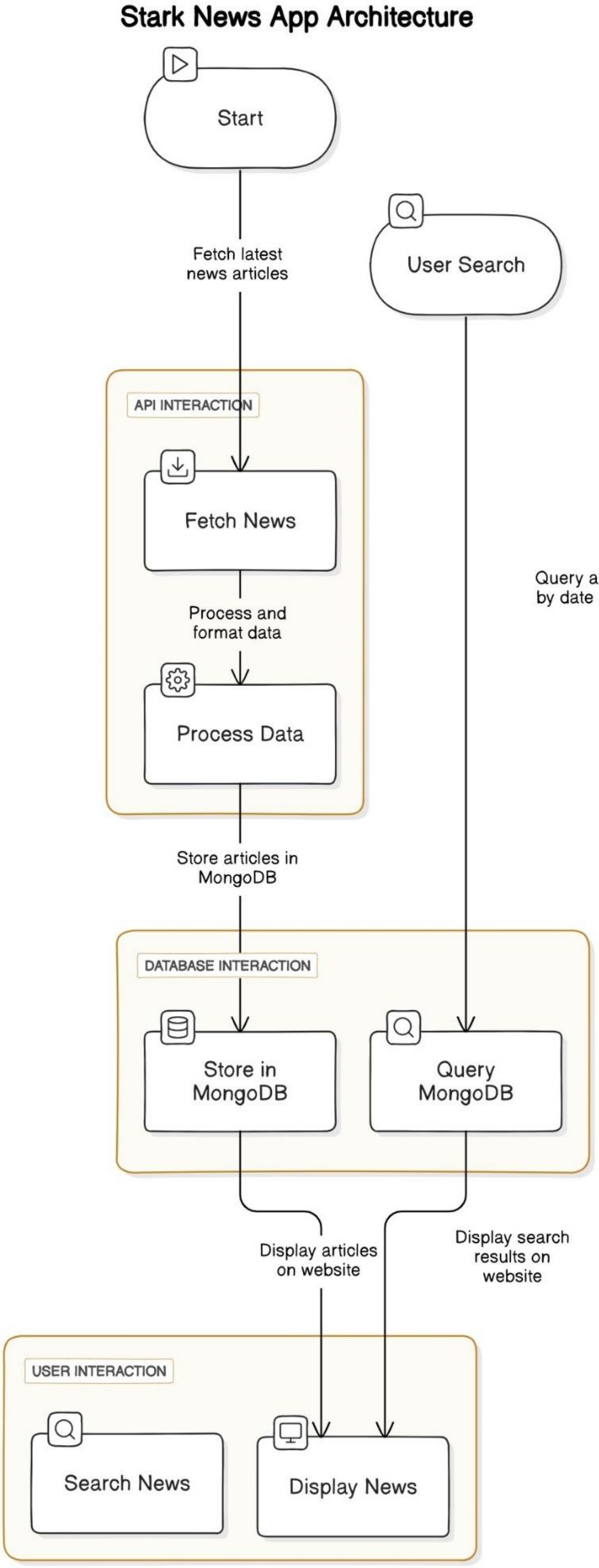
Software Requirements

- Operating System Windows 10
- Front End HTML , CSS , javascript
- Back End PYTHON, Postgre SQL

Hardware Requirements

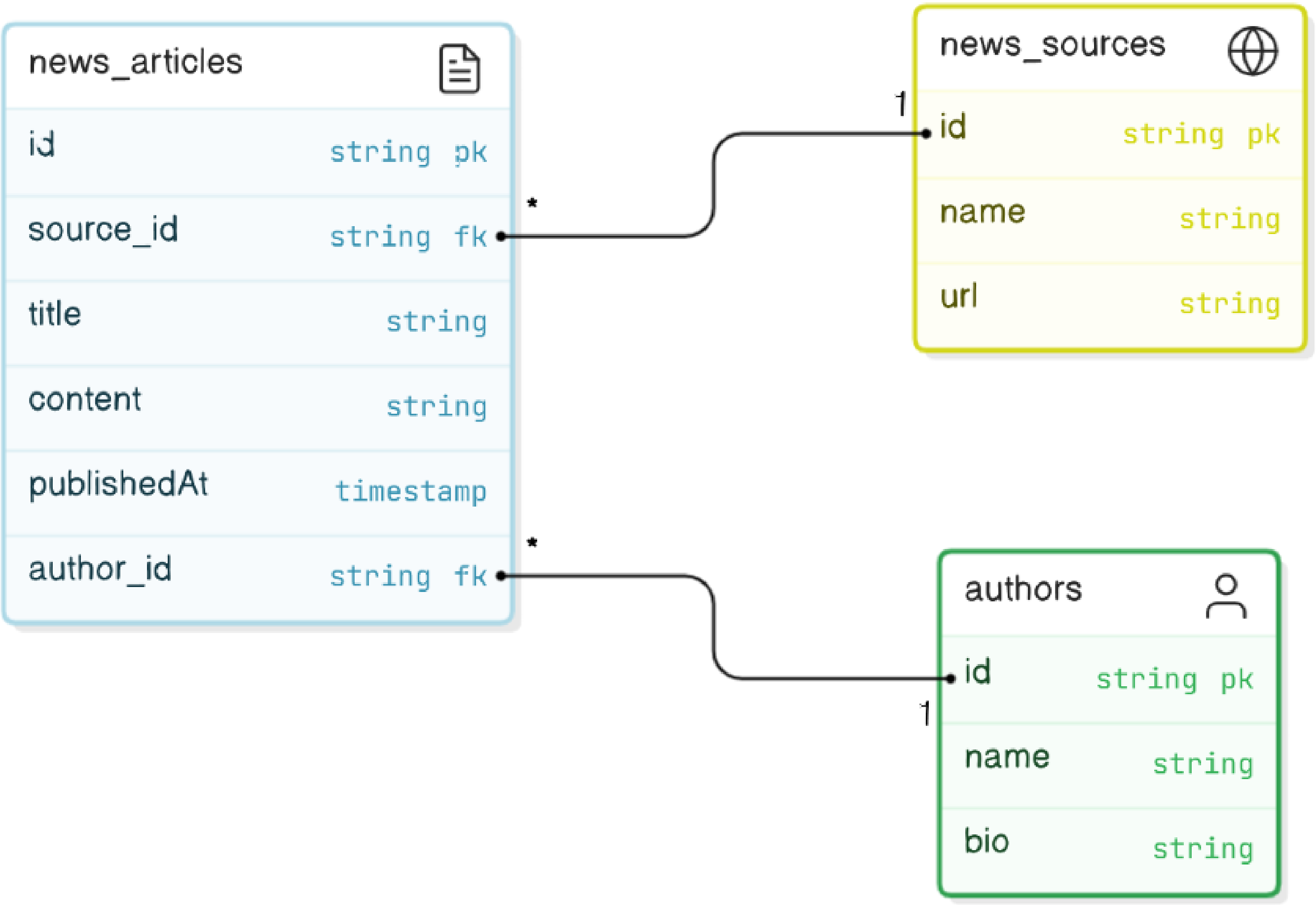
- Desktop PC or a Laptop
- Printer
- Operating System – Windows 10
- Intel® CoreTM i3-6006U CPU @ 2.00GHz ● 4.00 GB RAM
- 64-bit operating system, x64 based processor ● 1024 x 768 monitor resolution
- Keyboard and Mouse

3.3 ARCHITECTURE DIAGRAM


Stark News App Architecture
Start
Fetch latest news articles
User Search
API INTERACTION
Fetch News
Process and format data
Process Data
Query articles by date range
Store articles in MongoDB
DATABASE INTERACTION
Store in MongoDB
Query MongoDB
Display articles on website
Display search results on website
USER INTERACTION
Search News
Display News

3.4 ER-DIAGRAM

Stark News App ERD

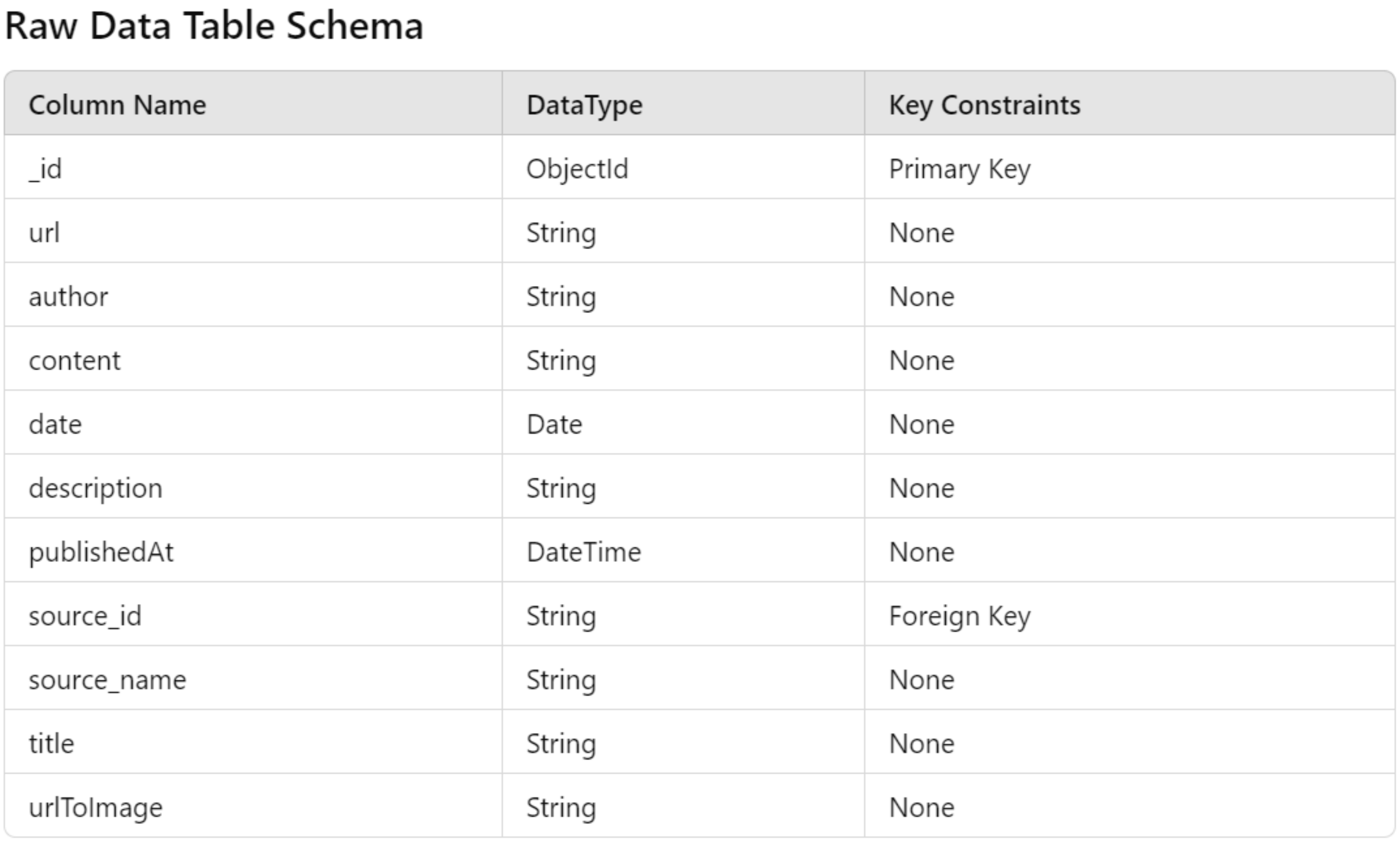


news_articles	
id	string pk
source_id	string fk
title	string
content	string
publishedAt	timestamp
author_id	string fk

news_sources	
id	string pk
name	string
url	string

authors	
id	string pk
name	string
bio	string

3.4 NORMALIZATION



Raw Data Table Schema

Column Name	DataType	Key Constraints
_id	ObjectId	Primary Key
url	String	None
author	String	None
content	String	None
date	Date	None
description	String	None
publishedAt	DateTime	None
source_id	String	Foreign Key
source_name	String	None
title	String	None
urlToImage	String	None

Normalized Tables

Articles Table

Column Name	DataType	Key Constraints
_id	ObjectId	Primary Key
url	String	None
author	String	None
content	String	None
date	Date	None
description	String	None
publishedAt	DateTime	None
source_id	String	Foreign Key
title	String	None
urlToImage	String	None

1NF

Articles Table

_id	url	author	content	date	description	publishedAt	source_id	title	urlT
ObjectId	String	String	String	Date	String	DateTime	String	String	Stri

2NF

Tables already meet the requirements of 2NF.

Ch 4. PROGRAM CODE

4.1 JARVISNEWSAPP

4.1.1 HTML/CSS AND JS

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>News Application</title>
    <link rel="icon" href="logo.png" type="image/png">
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            background-image: url('https://media.npr.org/assets/img/2013/05/06/tonyjarvis_wide-
92e2d9abcce4413d58f728f2b5f126cef71afd97.jpg');
            background-size: cover;
            background-repeat: no-repeat;
            background-attachment: fixed;
        }
        .container {
            width: 80%;
            margin: 20px auto;
            color: #ddd;
        }
        .selection {
            margin-bottom: 20px;
            display: flex;
            justify-content: space-between;
        }
        .selection form {
            display: inline-block;
        }
        .news-container {
            display: flex;
            flex-wrap: wrap;
            gap: 20px;
        }
        .news-card {
            background: #fff;
            border: 1px solid #ddd;
            border-radius: 5px;
            overflow: hidden;
            box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
            width: 300px;
            cursor: pointer;
        }
        .news-card img {
            width: 100%;
            height: auto;
        }
        .news-card-content {
            padding: 15px;
        }
        .news-card-title {
            font-size: 18px;
            margin: 0 0 10px;
            color: #0e0c0c;
        }
        .news-card-description {
            font-size: 14px;
            color: #0e0c0c;
        }
        .news-card a {
            text-decoration: none;
            color: inherit;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="selection">
            <form method="get" action="/">
                <label for="country">Select Country: </label>
                <select name="country" id="country" onchange="this.form.submit()">
                    <option value="ae" {% if country == 'ae' %}selected{% endif %}>United Arab Emirates</option>
<option value="ar" {% if country == 'ar' %}selected{% endif %}>Argentina</option>
<option value="at" {% if country == 'at' %}selected{% endif %}>Austria</option>
<option value="be" {% if country == 'be' %}selected{% endif %}>Belgium</option>
<option value="bg" {% if country == 'bg' %}selected{% endif %}>Bulgaria</option>
<option value="br" {% if country == 'br' %}selected{% endif %}>Brazil</option>
<option value="ch" {% if country == 'ch' %}selected{% endif %}>Switzerland</option>
<option value="cn" {% if country == 'cn' %}selected{% endif %}>China</option>
<option value="co" {% if country == 'co' %}selected{% endif %}>Colombia</option>
<option value="cu" {% if country == 'cu' %}selected{% endif %}>Cuba</option>
<option value="cz" {% if country == 'cz' %}selected{% endif %}>Czech Republic</option>
<option value="eg" {% if country == 'eg' %}selected{% endif %}>Egypt</option>
<option value="gr" {% if country == 'gr' %}selected{% endif %}>Greece</option>
<option value="hk" {% if country == 'hk' %}selected{% endif %}>Hong Kong</option>
<option value="hu" {% if country == 'hu' %}selected{% endif %}>Hungary</option>
<option value="id" {% if country == 'id' %}selected{% endif %}>Indonesia</option>
<option value="ie" {% if country == 'ie' %}selected{% endif %}>Ireland</option>
<option value="il" {% if country == 'il' %}selected{% endif %}>Israel</option>
<option value="in" {% if country == 'in' %}selected{% endif %}>India</option>
<option value="it" {% if country == 'it' %}selected{% endif %}>Italy</option>
<option value="jp" {% if country == 'jp' %}selected{% endif %}>Japan</option>
<option value="kr" {% if country == 'kr' %}selected{% endif %}>South Korea</option>
<option value="mx" {% if country == 'mx' %}selected{% endif %}>Mexico</option>
<option value="my" {% if country == 'my' %}selected{% endif %}>Malaysia</option>
<option value="ng" {% if country == 'ng' %}selected{% endif %}>Nigeria</option>
<option value="nl" {% if country == 'nl' %}selected{% endif %}>Netherlands</option>
<option value="no" {% if country == 'no' %}selected{% endif %}>Norway</option>
<option value="nz" {% if country == 'nz' %}selected{% endif %}>New Zealand</option>
<option value="ph" {% if country == 'ph' %}selected{% endif %}>Philippines</option>
<option value="pl" {% if country == 'pl' %}selected{% endif %}>Poland</option>
<option value="pt" {% if country == 'pt' %}selected{% endif %}>Portugal</option>
<option value="ro" {% if country == 'ro' %}selected{% endif %}>Romania</option>
<option value="ru" {% if country == 'ru' %}selected{% endif %}>Russia</option>
<option value="sa" {% if country == 'sa' %}selected{% endif %}>Saudi Arabia</option>
<option value="se" {% if country == 'se' %}selected{% endif %}>Sweden</option>
<option value="sg" {% if country == 'sg' %}selected{% endif %}>Singapore</option>
```

```
<option value="tr" {% if country == 'tr' %}selected{% endif %}>Turkey</option>
<option value="tw" {% if country == 'tw' %}selected{% endif %}>Taiwan</option>
<option value="ua" {% if country == 'ua' %}selected{% endif %}>Ukraine</option>
<option value="za" {% if country == 'za' %}selected{% endif %}>South Africa</option>

                </select>
            </form>
            <form method="get" action="/">
                <label for="category">Select Category: </label>
                <select name="category" id="category" onchange="this.form.submit()">
                    <option value="" {% if not category %}selected{% endif %}>None</option>
                    <option value="party" {% if category == 'party' %}selected{% endif
%}>Party</option>
                    <o<option value="" {% if not category %}selected{% endif %}>None</option>
                    <option value="party" {% if category == 'party' %}selected{% endif
%}>Party</option>
                    <option value="actor" {% if category == 'actor' %}selected{% endif
%}>Actor</option>
                    <option value="actress" {% if category == 'actress' %}selected{% endif
%}>Actress</option>
                    <option value="sports" {% if category == 'sports' %}selected{% endif
%}>Sports</option>
                    <option value="crime" {% if category == 'crime' %}selected{% endif
%}>Crime</option>
                    <option value="movies" {% if category == 'movies' %}selected{% endif
%}>Movies</option>
                    <option value="financial" {% if category == 'financial' %}selected{% endif
%}>Financial</option>
                    <option value="share market" {% if category == 'share market' %}selected{%
endif %}>Share Market</option>
                    <option value="gold" {% if category == 'gold' %}selected{% endif
%}>Gold</option>
                    <option value="marvel" {% if category == 'marvel' %}selected{% endif
%}>Marvel</option>
                    <option value="dceu" {% if category == 'dceu' %}selected{% endif
%}>DCEU</option>
                    <option value="anime" {% if category == 'anime' %}selected{% endif
%}>Anime</option>
                </select>
            </form>
        </div>
        <div class="news-container">
            {% for article in articles %}
            <div class="news-card">
                <a href="/article?title={{ article.title }}&image={{ article.urlToImage
}}&description={{ article.description }}&content={{ article.content }}">
                    <img src="{{ article.urlToImage }}" alt="News Image">
                    <div class="news-card-content">
                        <h3 class="news-card-title">{{ article.title }}</h3>
                        <p class="news-card-description">{{ article.description }}</p>
                    </div>
                </a>
            </div>
            {% endfor %}
        </div>
    </div>
</body>
</html>
```

ARTICLE VIEW

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{{ title }}</title>
    <link rel="icon" href="logo.png" type="image/png">
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            background-image:
url('https://media.npr.org/assets/img/2013/05/06/tonyjarvis_wide-
92e2d9abcce4413d58f728f2b5f126cef71afd97.jpg');
            background-size: cover;
            background-repeat: no-repeat;
            background-attachment: fixed;
        }

.container {
  width: 80%;
  margin: 20px auto;
  max-width: 1000px; /* Adjust for larger screens */
}

.news-article {
  background: #fff;
  border: 1px solid #ddd;
  border-radius: 5px;
  overflow: hidden;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  padding: 20px;
}

.news-article img {
  width: 100%;
  height: auto;
  margin-bottom: 15px; /* Add space between image and text */
}

.news-article-title {
  font-size: 24px;
  margin: 10px 0;
  font-weight: bold; /* Make title stand out */
}

.news-article-description {
  font-size: 18px;
  color: #555;
  margin: 10px 0;
  line-height: 1.5; /* Improve readability with spacing between lines */
}

.news-article-content {
  font-size: 16px;
  color: #333;
  margin: 10px 0;
}
    </style>
</head>
<body>
    <div class="container">
        <div class="news-article">
            <img src="{{ image_url }}" alt="News Image">
            <h1 class="news-article-title">{{ title }}</h1>
            <p class="news-article-description">{{ description }}</p>
            <p class="news-article-content">{{ content }}</p>
        </div>
    </div>
</body>
</html>
```

4.1.2 PYTHON

```
from flask import Flask, render_template, jsonify, request
import requests
from pymongo import MongoClient
from datetime import datetime

app = Flask(__name__)

# MongoDB setup
client = MongoClient('mongodb://localhost:27017/')
db = client.jarvis_news
news_collection = db.newsapp

def fetch_news(api_key, country='us', category=None):
    if category:
        url = f"https://newsapi.org/v2/everything?q={category}&apiKey={api_key}"
    else:
        url = f"https://newsapi.org/v2/top-headlines?country={country}&apiKey={api_key}"

    response = requests.get(url)
    news_data = response.json()

    if 'articles' in news_data:
        articles = news_data['articles']
        # Save articles to MongoDB with today's date
        for article in articles:
            article['country'] = country
            article['category'] = category
            article['date'] = datetime.today().strftime('%Y-%m-%d')
            news_collection.update_one(
                {'title': article['title'], 'date': article['date']},
                {'$set': article},
                upsert=True
            )
        return articles
    else:
        print(f"Error fetching news: {news_data}")
        return []

api_key = '3c2dc110a037439ab3eb00f313a9cd8e'

@app.route('/')
def home():
    country = request.args.get('country', 'us')
    category = request.args.get('category')
    news_articles = fetch_news(api_key, country, category)
    return render_template('ig.html', articles=news_articles, country=country, category=category)

@app.route('/article')
def article():
    title = request.args.get('title')
    image_url = request.args.get('image')
    description = request.args.get('description')
    content = request.args.get('content')
    return render_template('article.html', title=title, image_url=image_url,
description=description, content=content)

@app.route('/api/news')
def api_news():
    country = request.args.get('country', 'us')
    category = request.args.get('category')
    news_articles = fetch_news(api_key, country, category)
    return jsonify(news_articles)

if __name__ == '__main__':
    app.run(debug=True)
```

4.2 JARVISNEWSAPPSEARCH

4.2.1 html,css and js

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>News Articles for {{ date }}</title>
    <link rel="icon" href="logo.png" type="image/png">
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            background-image: url('https://media.npr.org/assets/img/2013/05/06/tonyjarvis_wide-
92e2d9abcce4413d58f728f2b5f126cef71afd97.jpg');
            background-size: cover;
            background-repeat: no-repeat;
            background-attachment: fixed;
        }
        .container {
            width: 80%;
            margin: 20px auto;
            color: #f4f4f4;
        }
        .selection {
            margin-bottom: 20px;
            display: flex;
            justify-content: space-between;
        }
        .selection form {
            display: inline-block;
        }
        .news-container {
            display: flex;
            flex-wrap: wrap;
            gap: 20px;
        }
        .news-card {
            background: #fff;
            border: 1px solid #ddd;
            border-radius: 5px;
            overflow: hidden;
            box-shadow: 0 2px 5px rgba(5, 5, 5, 0.1);
            width: 300px;
            cursor: pointer;
        }
        .news-card img {
            width: 100%;
            height: auto;
        }
        .news-card-content {
            padding: 15px;
        }
        .news-card-title {
            font-size: 18px;
            margin: 0 0 10px;
            color: black;
        }
        .news-card-description {
            font-size: 14px;
            color: #111010;
        }
        .news-card a {
            text-decoration: none;
            color: inherit;
        }
        .back-link {
            display: block;
            margin-top: 20px;
            text-align: center;
            color: #007BFF;
            text-decoration: none;
        }
        .back-link:hover {
            text-decoration: underline;
        }
        .error {
            color: red;
            margin-bottom: 20px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Search News Articles</h1>
        <form method="POST" action="/">
            <label for="date">Enter Date (YYYY-MM-DD):</label>
            <input type="text" id="date" name="date" required>
            <button type="submit">Search</button>
        </form>
        {% if error %}
            <p class="error">{{ error }}</p>
        {% endif %}
        {% if articles is not none %}
            <h1>News Articles for {{ date }}</h1>
            <div class="news-container">
                {% if articles %}
                    {% for article in articles %}
                        <div class="news-card">
                            {% if article.urlToImage %}
                                <img src="{{ article.urlToImage }}" alt="Article Image">
                            {% endif %}
                            <div class="news-card-content">
                                <h2 class="news-card-title">{{ article.title }}</h2>
                                <p class="news-card-description">{{ article.description }}</p>
                                <a href="{{ article.url }}" target="_blank">Read more</a>
                            </div>
                        </div>
                    {% endfor %}
                {% else %}
                    <p>No articles found for this date.</p>
                {% endif %}
            </div>
            <a href="/" class="back-link">Back to Search</a>
        {% endif %}
    </div>
</body>
</html>
```

4.2.2 PYTHON

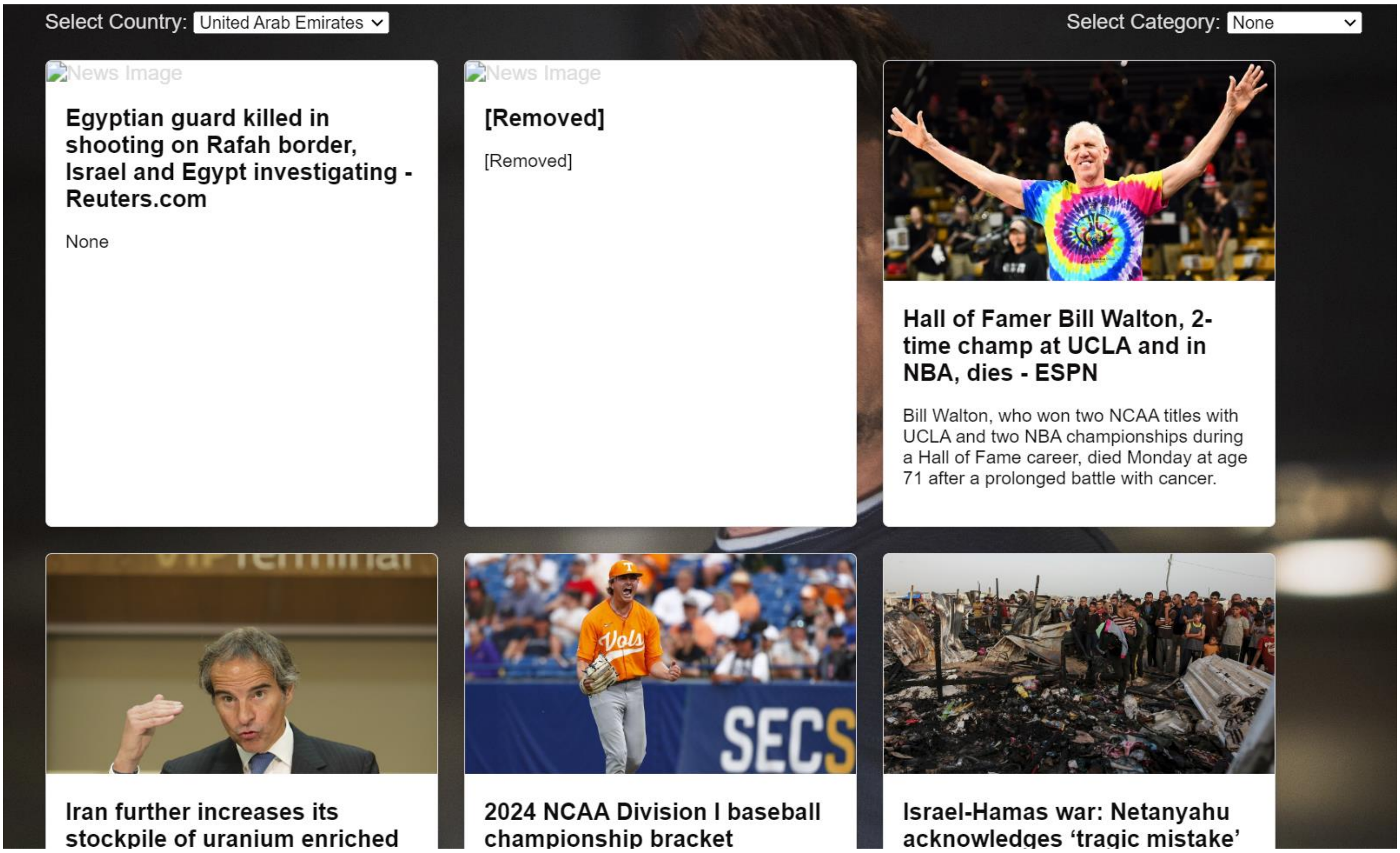
```python
from flask import Flask, render_template, request
from pymongo import MongoClient
from datetime import datetime

app = Flask(__name__)

client = MongoClient('mongodb://localhost:27017/')
db = client.jarvis_news
news_collection = db.newsapp

@app.route('/', methods=['GET', 'POST'])
def index():
    articles = None
    date_str = None
    if request.method == 'POST':
        date_str = request.form.get('date')
        if not date_str:
            return render_template('index.html', error="Please provide a date in the format YYYY-MM-DD")

        try:
            query_date = datetime.strptime(date_str, '%Y-%m-%d').date()
        except ValueError:
            return render_template('index.html', error="Invalid date format. Please use YYYY-MM-DD")

        # Query the collection for articles matching the date
        articles = news_collection.find({"date": str(query_date)})
        # Convert articles to a list
        articles = list(articles)

    return render_template('index.html', articles=articles, date=date_str)

if __name__ == '__main__':
    app.run(debug=True)
```

Ch 5. RESULTS AND DISCUSSION

5.1 USER DOCUMENTATION

JARVISNEWSAPP

COUNTRY NEWS



Select Country: United Arab Emirates

Select Category: None

News Image

Egyptian guard killed in shooting on Rafah border, Israel and Egypt investigating - Reuters.com

None

News Image

[Removed]

[Removed]

Hall of Famer Bill Walton, 2-time champ at UCLA and in NBA, dies - ESPN

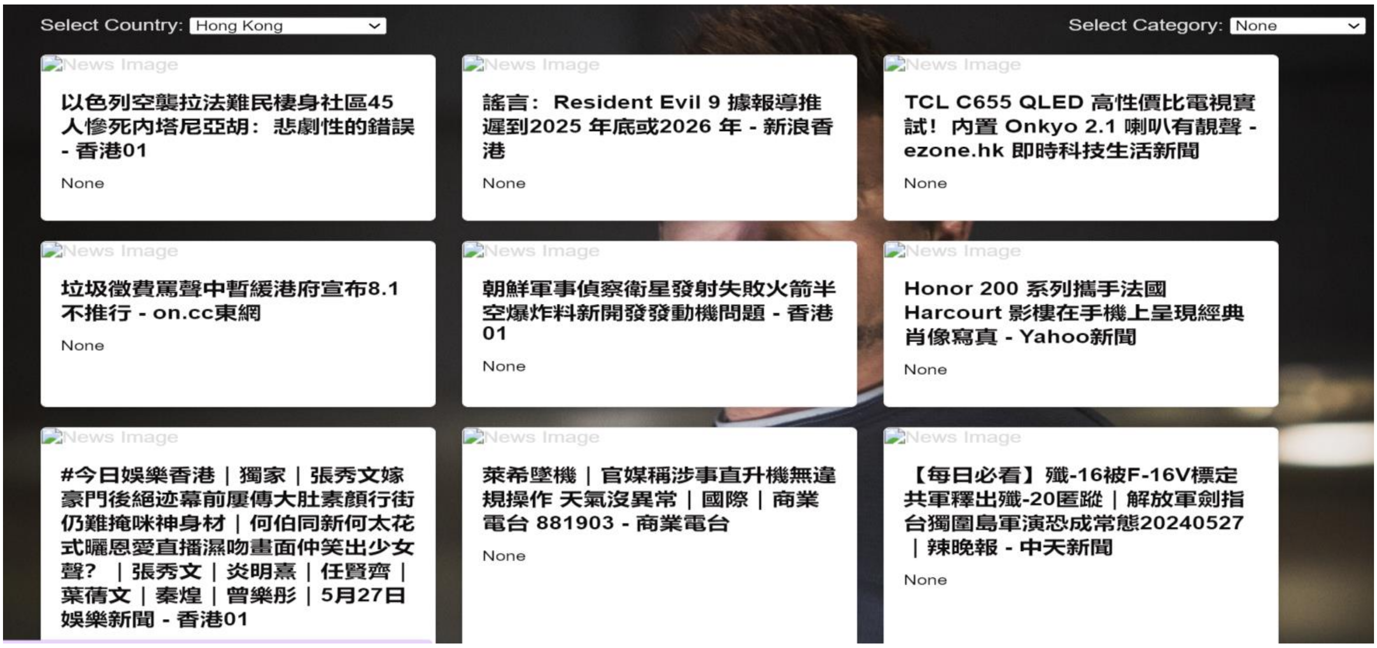
Bill Walton, who won two NCAA titles with UCLA and two NBA championships during a Hall of Fame career, died Monday at age 71 after a prolonged battle with cancer.

Iran further increases its stockpile of uranium enriched

2024 NCAA Division I baseball championship bracket

Israel-Hamas war: Netanyahu acknowledges ‘tragic mistake’

COUNTRY NEWS 40+ LANGAUGES



Select Country: Hong Kong

Select Category: None

News Image

以色列空襲拉法難民棲身社區45人慘死內塔尼亞胡：悲劇性的錯誤 - 香港01

None

News Image

謠言： Resident Evil 9 據報導推遲到2025 年底或2026 年 - 新浪香港

None

News Image

TCL C655 QLED 高性價比電視實試！ 內置 Onkyo 2.1 喇叭有靚聲 - ezone.hk 即時科技生活新聞

None

News Image

垃圾徵費罵聲中暫緩港府宣布8.1不推行 - on.cc東網

None

News Image

朝鮮軍事偵察衛星發射失敗火箭半空爆炸料新開發發動機問題 - 香港01

None

News Image

Honor 200 系列攜手法國 Harcourt 影樓在手機上呈現經典肖像寫真 - Yahoo新聞

None

News Image

#今日娛樂香港｜獨家｜張秀文嫁豪門後絕迹幕前屢傳大肚素顏行街仍難掩咪神身材｜何伯同新何太花式曬恩愛直播濕吻畫面仲笑出少女聲？ ｜張秀文｜炎明熹｜任賢齊｜葉蒨文｜秦煌｜曾樂彤｜5月27日娛樂新聞 - 香港01

News Image

萊希墜機｜官媒稱涉事直升機無違規操作 天氣沒異常｜國際｜商業電台 881903 - 商業電台

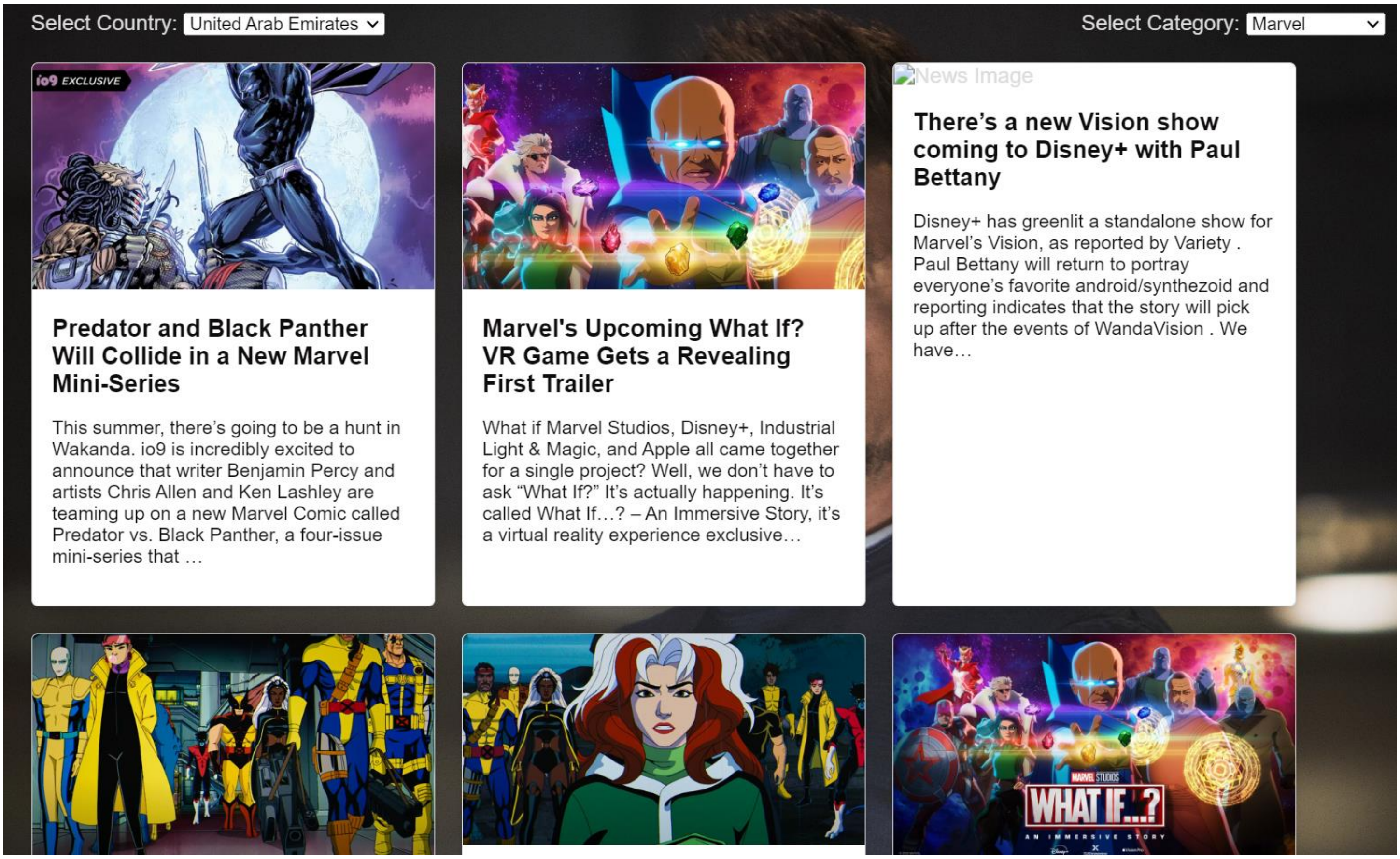
None

News Image

【每日必看】殲-16被F-16V標定 共軍釋出殲-20匿蹤｜解放軍劍指台獨圍島軍演恐成常態20240527｜辣晚報 - 中天新聞

None

CATEGORY NEWS



Select Country: United Arab Emirates

Select Category: Marvel

Predator and Black Panther Will Collide in a New Marvel Mini-Series

This summer, there's going to be a hunt in Wakanda. io9 is incredibly excited to announce that writer Benjamin Percy and artists Chris Allen and Ken Lashley are teaming up on a new Marvel Comic called Predator vs. Black Panther, a four-issue mini-series that …

Marvel's Upcoming What If? VR Game Gets a Revealing First Trailer

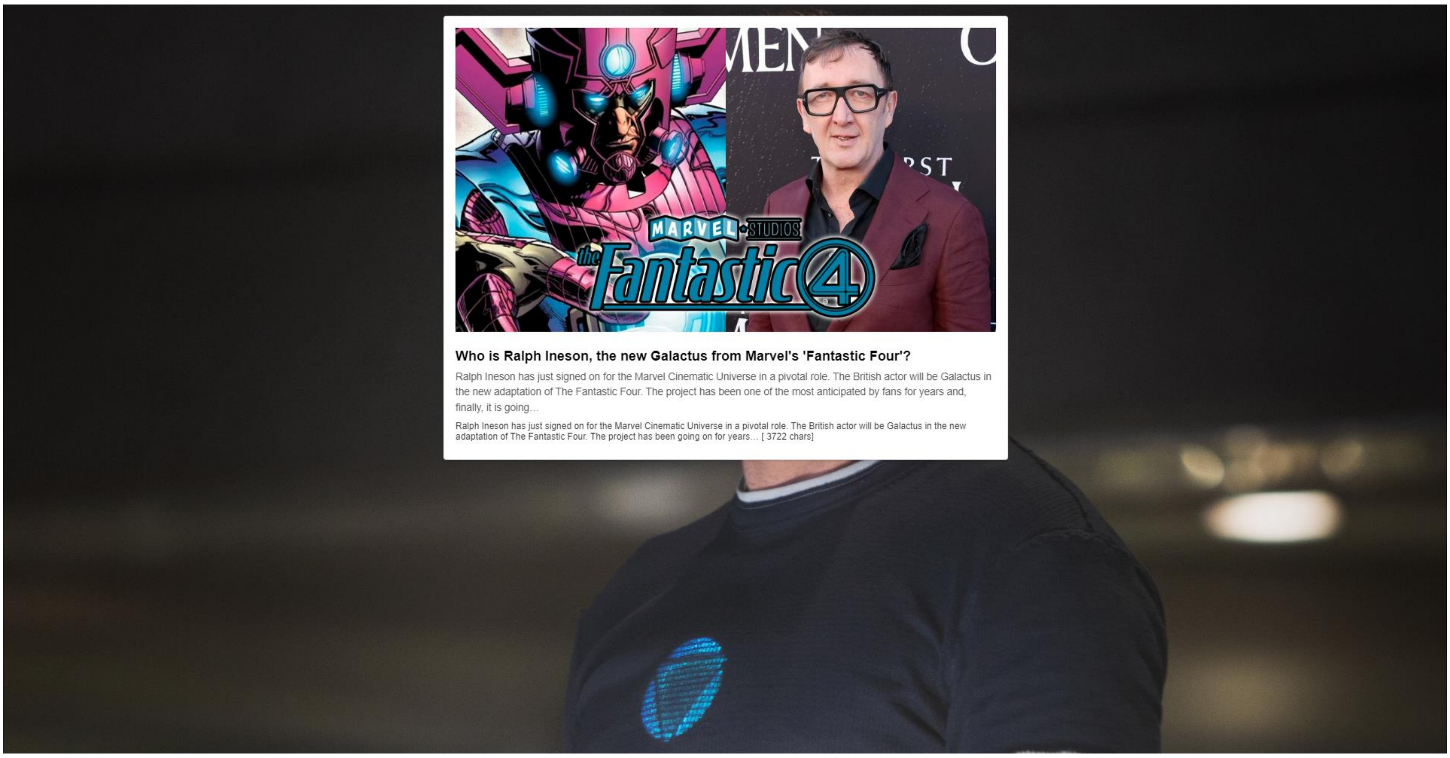
What if Marvel Studios, Disney+, Industrial Light & Magic, and Apple all came together for a single project? Well, we don't have to ask "What If?" It's actually happening. It's called What If…? – An Immersive Story, it's a virtual reality experience exclusive…

News Image

There's a new Vision show coming to Disney+ with Paul Bettany

Disney+ has greenlit a standalone show for Marvel's Vision, as reported by Variety . Paul Bettany will return to portray everyone's favorite android/synthezoid and reporting indicates that the story will pick up after the events of WandaVision . We have…

DETAIL ARTICLE VIEWER

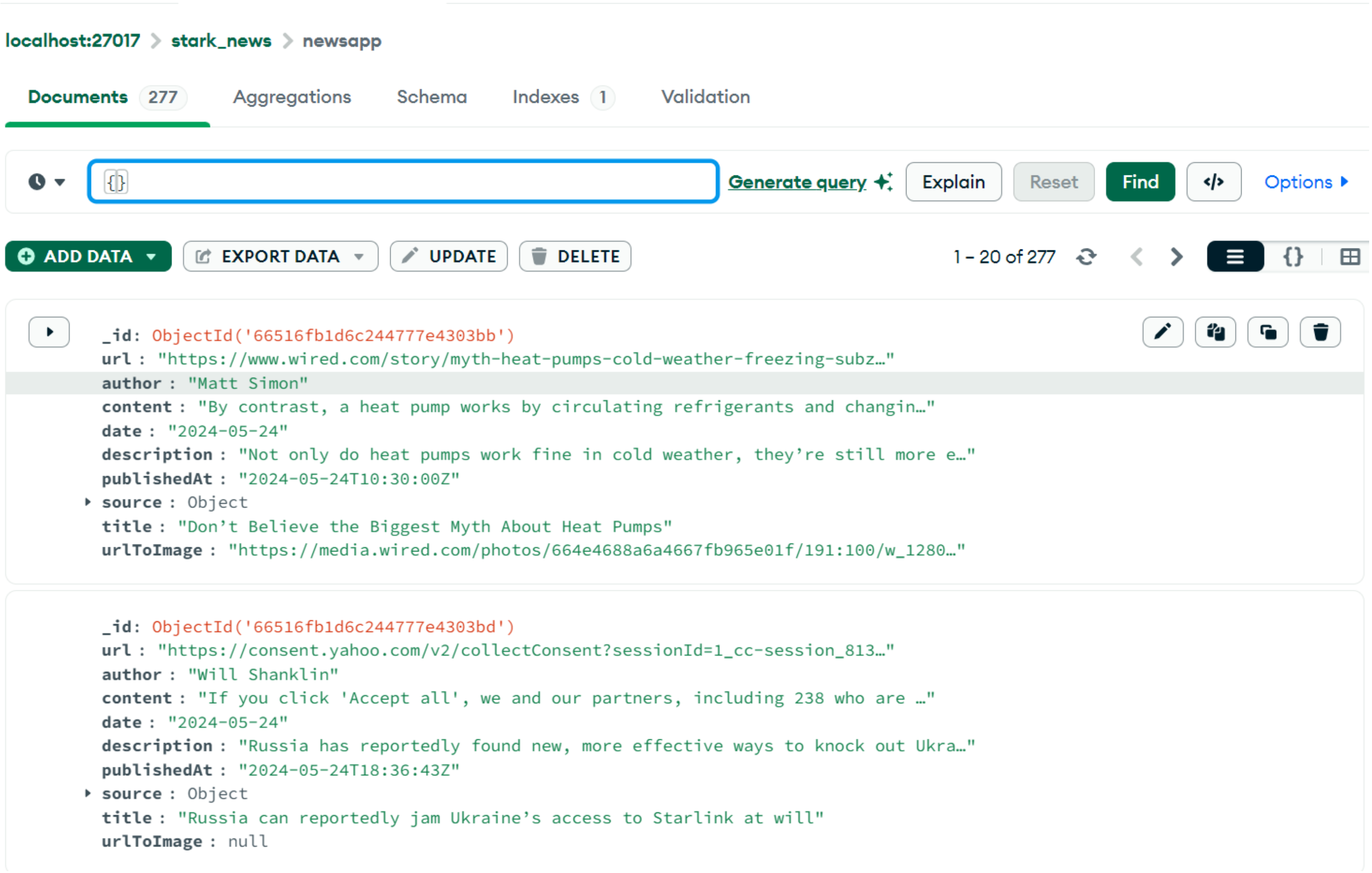


Who is Ralph Ineson, the new Galactus from Marvel's 'Fantastic Four'?

Ralph Ineson has just signed on for the Marvel Cinematic Universe in a pivotal role. The British actor will be Galactus in the new adaptation of The Fantastic Four. The project has been one of the most anticipated by fans for years and, finally, it is going…

Ralph Ineson has just signed on for the Marvel Cinematic Universe in a pivotal role. The British actor will be Galactus in the new adaptation of The Fantastic Four. The project has been going on for years… [3722 chars]

MONGODB DATABASE VIEW

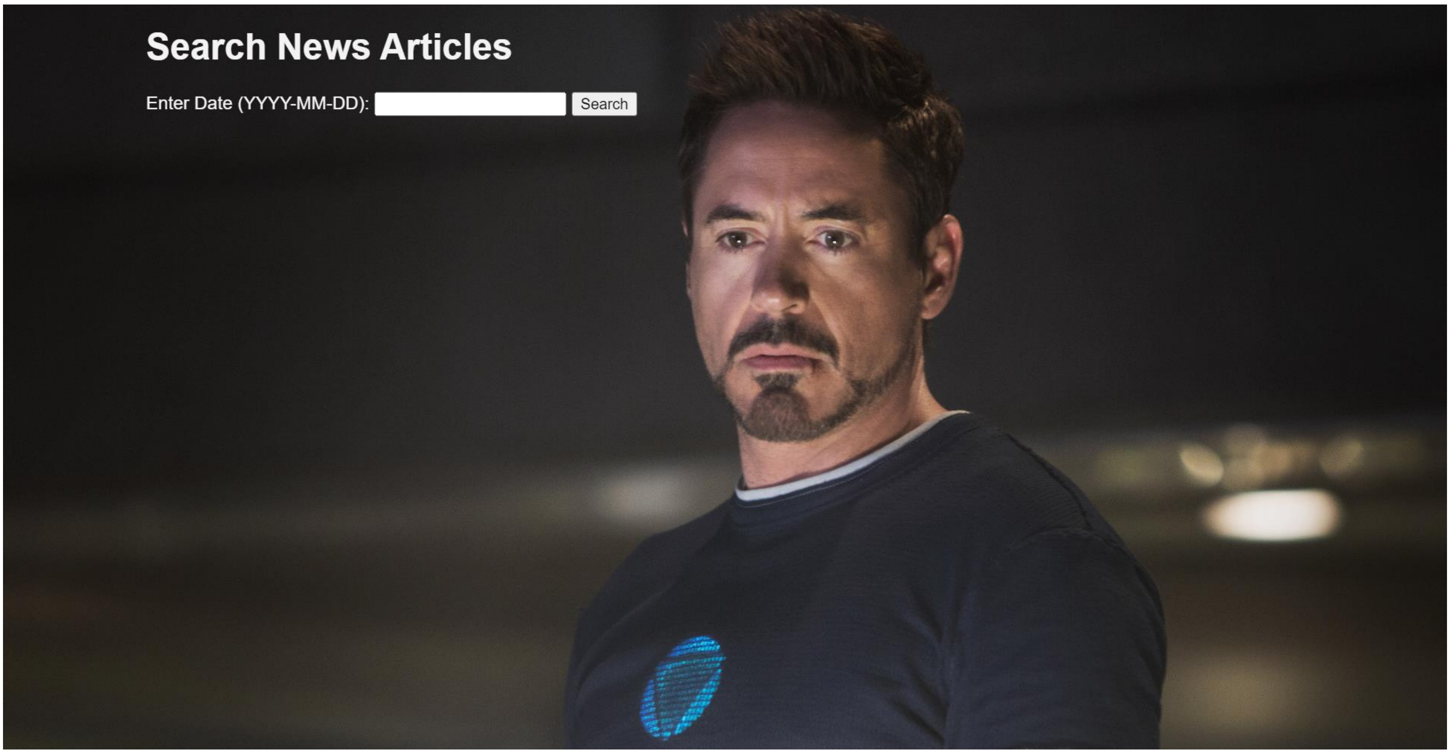


```
_id: ObjectId('66516fb1d6c244777e4303bb')
url : "https://www.wired.com/story/myth-heat-pumps-cold-weather-freezing-subz…"
author : "Matt Simon"
content : "By contrast, a heat pump works by circulating refrigerants and changin…"
date : "2024-05-24"
description : "Not only do heat pumps work fine in cold weather, they’re still more e…"
publishedAt : "2024-05-24T10:30:00Z"
source : Object
title : "Don’t Believe the Biggest Myth About Heat Pumps"
urlToImage : "https://media.wired.com/photos/664e4688a6a4667fb965e01f/191:100/w_1280…"
```

```
_id: ObjectId('66516fb1d6c244777e4303bd')
url : "https://consent.yahoo.com/v2/collectConsent?sessionId=1_cc-session_813…"
author : "Will Shanklin"
content : "If you click 'Accept all', we and our partners, including 238 who are …"
date : "2024-05-24"
description : "Russia has reportedly found new, more effective ways to knock out Ukra…"
publishedAt : "2024-05-24T18:36:43Z"
source : Object
title : "Russia can reportedly jam Ukraine’s access to Starlink at will"
urlToImage : null
```

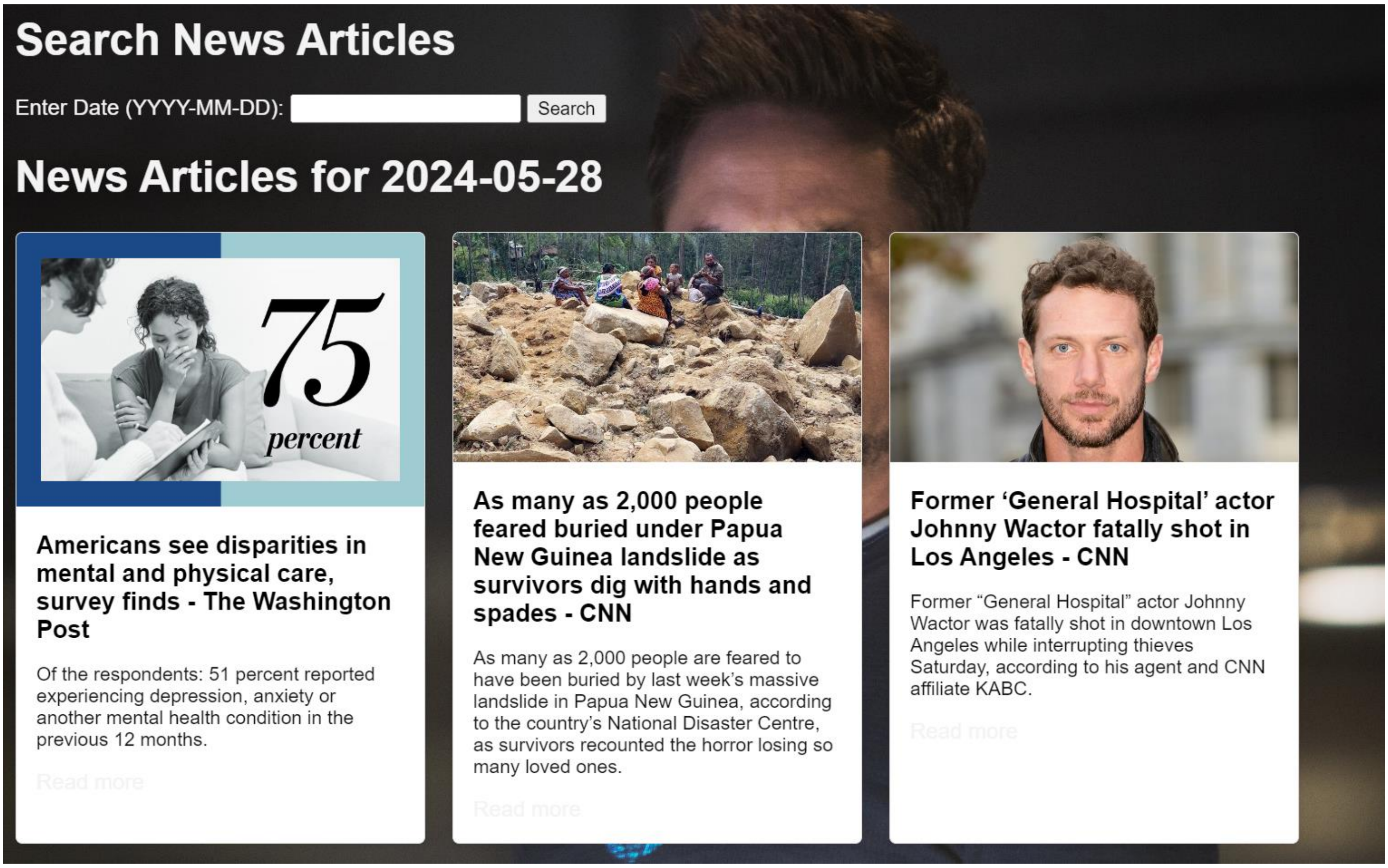
SEARCH OF PREVIOUS

PAGE



Search News Articles

SEARCH VIEW



Search News Articles

News Articles for 2024-05-28

Americans see disparities in mental and physical care, survey finds - The Washington Post

Of the respondents: 51 percent reported experiencing depression, anxiety or another mental health condition in the previous 12 months.

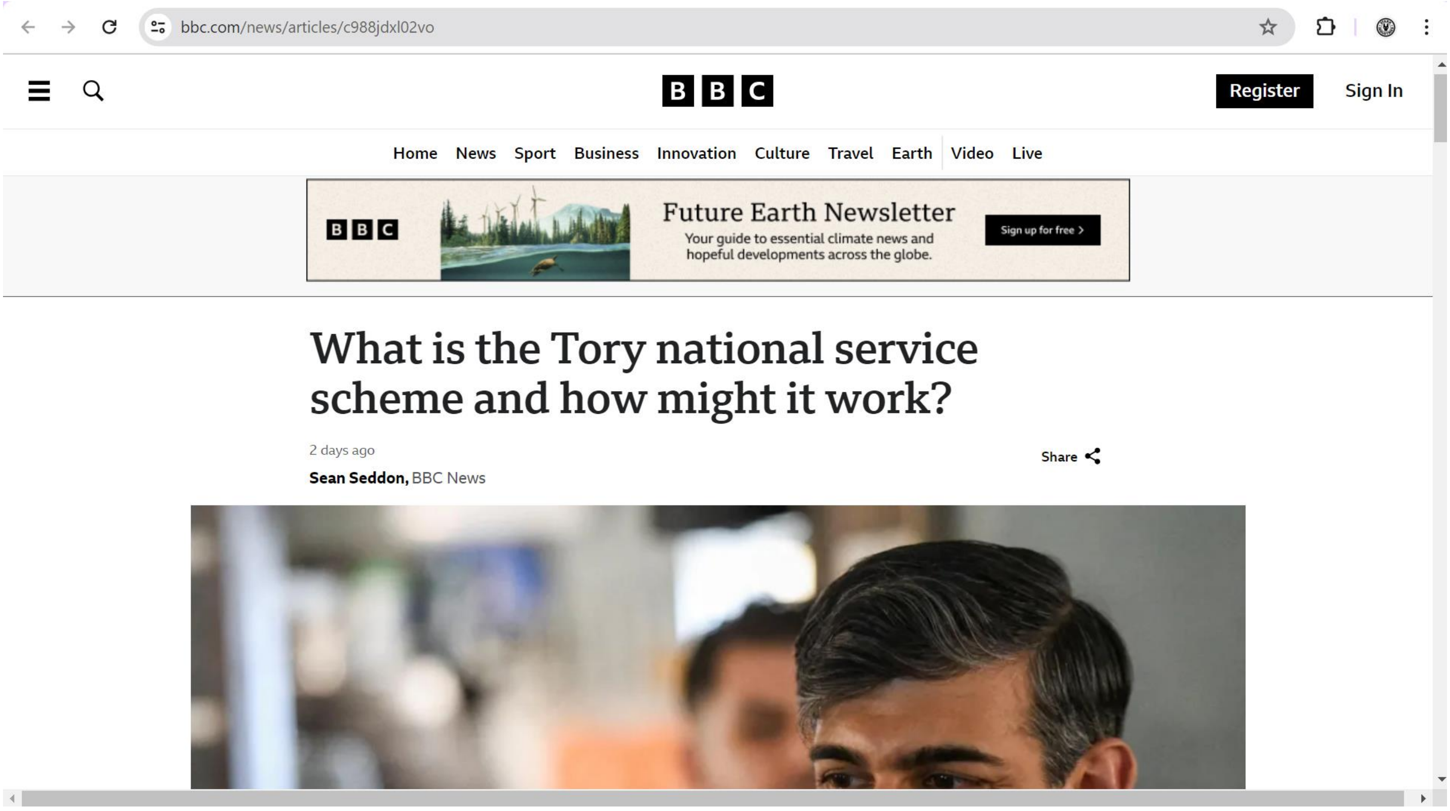
As many as 2,000 people feared buried under Papua New Guinea landslide as survivors dig with hands and spades - CNN

As many as 2,000 people are feared to have been buried by last week's massive landslide in Papua New Guinea, according to the country's National Disaster Centre, as survivors recounted the horror losing so many loved ones.

Former 'General Hospital' actor Johnny Wactor fatally shot in Los Angeles - CNN

Former "General Hospital" actor Johnny Wactor was fatally shot in downtown Los Angeles while interrupting thieves Saturday, according to his agent and CNN affiliate KABC.

ONLINE LINK VIEW



bbc.com/news/articles/c988jdxl02vo

What is the Tory national service scheme and how might it work?

2 days ago

Sean Seddon, BBC News

5.2 DISCUSSION

- **Scalability**: The architecture is designed to handle large volumes of news articles efficiently. By utilizing MongoDB as the database, the app can scale horizontally to accommodate increasing data loads. This ensures that as the number of users and news articles grows, the system remains responsive and performs well.

- **Flexibility**: The integration with the News API allows for flexibility in sourcing news content. Users can specify various parameters such as sources, categories, or keywords to tailor their news feed according to their interests. This flexibility enhances the user experience by providing personalized content.

- **Reliability**: The use of MongoDB as the backend database contributes to the reliability of the system. MongoDB's robust replication and failover mechanisms ensure high availability and data durability. This reliability is crucial for ensuring that users can access news articles consistently without interruptions.

- **Search Functionality**: The ability to search for news articles by date adds value to the app. Users can easily retrieve past news articles based on their specific interests or events they want to revisit. This feature enhances the usability of the app and improves user satisfaction.

- **Data Management**: MongoDB's document-oriented nature simplifies the storage and retrieval of news articles. Each article is stored as a JSON document, allowing for flexible schema design and efficient querying. This enables fast retrieval of news articles based on various criteria, enhancing the overall performance of the app.

Ch 6. CONCLUSION

In conclusion, the architecture of the Jarvis News app embodies a robust and efficient system for fetching, storing, and displaying news articles. By integrating the News API for sourcing real-time news content and MongoDB for storing articles, the app offers scalability, flexibility, and reliability.

The app's ability to scale horizontally ensures it can handle increasing data loads as user traffic and news volumes grow. Additionally, the flexibility provided by the News API allows users to personalize their news feed according to their interests, enhancing the overall user experience.

The inclusion of a search functionality based on date enables users to retrieve past news articles easily, adding value to the app and improving user satisfaction. MongoDB's document-oriented approach simplifies data management, facilitating fast and efficient storage and retrieval of news articles.

Overall, the architecture of the Jarvis News app is well-designed to meet the demands of a modern news consumption platform. It provides a seamless user experience, ensuring reliable access to up-to-date news content while offering flexibility and scalability for future growth and development.

Ch 7. REFERENCE

- MONGODB DATABASE CONNECTIVITY BY GEEKS OF GEEKS
- NEWS APP CREATOR ZOHO CREATORS