

# **NEWS TRACKER APPPLICATION**

USING CLOUD

*A Project report submitted in partial fulfilment of 7<sup>th</sup> semester in degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

*Submitted by*

Team ID: PNT2022TMID51998

Yogesh P 962319104103

Sridevi M I 962319104089

Mahalingam M 962319104055

Sunil I 962319104092

Vishnu Murugan 962319104102



**DEPARTMENT OF COMPUTER SCIENCEAND ENGINEERING**

**AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**ANNA UNIVERSITY: CHENNAI 600025**

**NOV-2022**

# **AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(A Constituent College of Anna University, Chennai)**



## **BONAFIDE CERTIFICATE**

Certified that this project report "NEWS TRACKER APPLICATION" is the bonafide record work done by **Mr Yogesh P (962319104103)**, **Ms Sridevi M I (962319104089)**, **Mr Mahalingam M (962319104055)**, **Mr Sunil I (962319104092)** and **Mr Vishnu Murugan (962319104102)** for **IBM-NALAIYATHIRAN** in **VII** semester of **B.E.**, degree course in **Computer Science and Engineering** branch during the academic year of 2022 - 2023.

**Staff-Incharge**

**Mr. I. Anantraj**

**Evaluator**

**Mrs K. Rejini**

**Head of the Department**

**Dr. P M Siva Raja**

## **ACKNOWLEDGEMENT**

We express our breathless thanks to our **Dr.T.Kannan, M.E., Ph.D.**, the principal, Amrita College of Engineering and Technology, Erachakulam, for giving constant motivation in succeeding in our goal.

We acknowledge our sincere thanks to Head of the Department (i/c) **Dr. P M Siva Raja M.Tech., Ph.D.**, for giving us valuable suggestion and help towards us throughout this Project.

We are highly grateful to thank our Project coordinator **Mr. I. Anantraj M.E., MBA , (Ph.D.)**, and our Project Evaluator **Mrs K. Rejini M.E., (Ph.D.)**, Department of Computer Science and Engineering, Amrita College of Engineering and technology, for the coordinating us throughout this Project.

We are very much indebted to thank all the faculty members of Department of Computer science and Engineering in our Institute, for their excellent moral support and suggestions to complete our Project work successfully.

Finally our acknowledgment goes to our parents, sisters and friends those who had extended their excellent support and ideas to make our Project a pledge one.

**Yogesh P  
Sridevi M I  
Mahalingam M  
Sunil I  
Vishnu Murugan**

## **TABLE OF FIGURES**

### **1. INTRODUCTION**

- a. Project Overview
- b. Purpose

### **2. LITERATURE SURVEY**

- a. Existing problem
- b. References
- c. Problem StatementDefinition

### **3. IDEATION &PROPOSED SOLUTION**

- a. Empathy Map Canvas
- b. Ideation & Brainstorming
- c. Proposed Solution
- d. Problem Solutionfit

### **4. REQUIREMENT ANALYSIS**

- a. Functional requirement
- b. Non-Functional requirements

### **5. PROJECT DESIGN**

- a. Data Flow Diagrams
- b. Solution & Technical Architecture
- c. User Stories

### **6. PROJECT PLANNING& SCHEDULING**

- a. Sprint Planning & Estimation
- b. Sprint DeliverySchedule
- c. Reports from JIRA

### **7. CODING & SOLUTIONING**

- a. Feature 1
- b. Feature 2
- c. Database Schema (if Applicable)

## **8. TESTING**

- a. Test Cases
- b. User Acceptance Testing

## **9. RESULTS**

- a. Performance Metrics

## **10.ADVANTAGES & DISADVANTAGES**

## **11.CONCLUSION**

## **12.FUTURE SCOPE**

## **13.APPENDIX**

Source Code GitHub& Project Demo Link

# News Tracker Application

## 1.INTRODUCTION :

Web app ecosystems are transforming patterns of news consumption. Until quite recently, reading the news was a niche use for smartphones , mostly for when users were ‘on the go’; now however, two in every three users of mobile devices / web applications in the US regularly access news and as many as one in five read in-depth news articles daily ; a similar picture is found in the UK . This growth in web news access continues the migration of news consumers to the Internet.

### 1.1 Project overview

As news is increasingly accessed on smartphones and tablets, the need for personalising news app interactions is apparent. We report a series of three studies addressing key issues in the development of adaptive news app interfaces. We first surveyed users’ news reading preferences and behaviours; analysis revealed three primary types of reader. We then implemented and deployed an Android news app that logs users’ interactions with the app. We used the logs to train a classifier and showed that it is able to reliably recognise a user according to their readertype. Finally we evaluated alternative, adaptive user interfaces for each reader type. The evaluation demonstrates the differential benefit of the adaptation for different users of the news app and the feasibility of adaptive interfaces for news apps.

### 1.2 Purpose

As our lives are very busy these days, we often feel we need more than 24 hrs. a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you're interested in and get a quick peek for the day. Only read what you feel is relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Future Rates, Bonds, etc.... as on official websites.

## 2.LITERATURE SURVEY

### 2.1 Existing problem

We often feel we need more than 24 hrs. a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help.

### 2.2 Literature Survey

SI No	Title	Authors	Year	Technique	Merits	Demer
	Design and Implementation of News Collecting and Filtering System Based on RSS	Zheng, R., & Zhang, Y.	2012	Using RSS to collect news with enhanced search system	System can automatically collect the latest news information from the subscribe site, then parsing and Storing the information into database.	Graphics and photos do not always appear .Posts are easily de-formatted or fully erased. RSS might make the process a bit unpleasant.
	News Event Detection and Tracking Based on Stream of Online News	<a href="#">Yajie Qi</a> <a href="#">Li Zhou</a> <a href="#">Huayou Si</a> <a href="#">JianWan</a> <a href="#">Ting Jin</a>		Single-pass clustering algorithm for event detection and tracking	Extraction of news content on particular field by searching keywords.	In some cases it can't analyse the keyword brings original news or not
	Deep News	Kong, X., Kong, Q.,		Word embedding	Top news will be ranked according to the user query	The model is trained on the co-occurrence matrix of words,

	<b>Event Ranker Based on User Relevant Query</b>	Mao, W., & Tang, S.	2018	technology using Global vector of word representation		which takes a lot of memory for storage
<b>4</b>	<b>Exploring mobile news reading interactions for news app personalisation</b>	Marios Constantini de s, John Dowell, David Johson, Sylvain Malacria	2015	1. Identification of news reader types 2. Interaction logging and classification study 3. Deployment and data collection 4. Predicting News reader types 5. Adaptive UI	The adaptive user interface changes according to the type of task you want to perform. This will increase the stability of the system.	The overall code of the website and the size of the app increases if the user interface is adaptive. There is a lot of code to be written for making the user interface adaptive
<b>5</b>	<b>Detection and Tracking in News Articles</b>	Sagar Patel, Sanket Suthar, Sandip Patel, Neha Patel	2015	1. Pre-processing 2. Tokenization 3. Stemming/Lemmatization 4. Vector Space Model 5. Topic tracking	Allows the computing for a continuous degree of similarities between queries and document	1. Suffers from synonym and polysemy 2. It theoretically assumes that terms are statistically independent.
<b>6</b>	<b>Following the Fed with a News Tracker</b>	Michael William McCracken	2012	The paper is not a technical paper but is essentially a statistical paper on how should one	Tracks whether a core set of economic data series has been coming in under expectations, at expectations, or	The surprise indexes tend to have the right sign and be significant: a positive change in the U.S.



				conclude whether the data have come in stronger, weaker or as expected. This is based on the Citi Group U.S Economic Surprise Index.	over expectations	surprise index (i.e. the U.S. economy doing better than expected) appreciates the U.S. dollar versus the foreign currency, whereas a positive change in the foreign surprise index depreciated the U.S. dollar.
--	--	--	--	--	-------------------	---

## 2.3 Problem StatementDefinition

1. The user needs a way to organize the news on daily basis so that he can cope up with daily events in his tight schedule.
2. The user wants to read news only about particular topics so that he can be informed about his interest.
3. The user wants to get informed from only certified news outlets.
4. The user needs a way to search about the news on topics he wants to.
5. The user wants to know the news about his surrounding using GPS location.

### **3. IDEATION & PROPOSED SOLUTION**

#### **3.1 Empathy Map Canvas**

#### **3.2 Ideation & Brainstorming**

### 3.3 Proposed Solution

S.NO	PARAMETERS	DESCRIPTION
1.	Problem statement (problems to be solved)	<p>As our lives are very busy these days, we often feel we need more than 24 hrs. a day to cope up with everything we have in our schedule. Well, that's not possible but reducing the time by changing the conventional method of reading news can help. Just tell us what market news you're interested in and get a quick peek for the day. Only read what you feel is relevant and save your time. This app helps you to query for all information about Indices, Commodities, Currencies, Future Rates, Bonds, etc.... as on official websites.</p>
2.	Idea/solution description	<p>We encounter several events that are significant locally and nationally in our daily lives, and we use NEWS to learn about them. But we recently came to a realization that there isn't a single platform that offers both. Thus, after identifying a social need, we developed a platform that offers such news in the form of brief videos that are gathered from the most reliable news sources worldwide. What we will do :</p> <ul style="list-style-type: none"><li>· We deliver news in-depth and accurately.</li><li>· We will be delivering information from reliable, reliable sources worldwide to prevent the spread of fake news</li><li>· We also offer local news coverage based on user location</li><li>· All of his favourite local news sources could be found on one platform.</li><li>· We have also multilingualized our platform so that locals could readily understand it</li></ul>

3.	Novelty/Uniqueness	<ul style="list-style-type: none"> <li>· The application offers language translation</li> <li>· Real-time reporting is required for news which is also provided.</li> </ul>
4.	Social Impact/Customer satisfaction	<ul style="list-style-type: none"> <li>· Users will find the recommendations interesting and helpful because news is suggested based on their interests And past behavior..</li> </ul>
5.	Business Model(Revenue)	Through hourly news subscriptions, we can generate revenue.
6.	Scalability of the solution	This application is totally constructed using a scalable architecture. Therefore, this application can be expanded to meet any amount of traffic, making the user experience seamless regardless of the Increase in demand. This application is totally hosted in the cloud, so when demand increases, the configurations and processing Power can be changed to give users a seamless experience

### 3.3 Problem Solution fit

## 4.REQUIREMENT ANALYSIS

### 4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement(Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	User using his /her device to login to the app/website for reading news
FR-4	User review	It helps to develop the app further and improve our application features.

### 4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
--------	----------------------------	-------------

NFR-1	<b>Usability</b>	Need to update their knowledge towards their interested news.
NFR-2	<b>Security</b>	The information collected from user would be very safe and secure.
NFR-3	<b>Reliability</b>	It will operate in a defined environment without failure.
NFR-4	<b>Performance</b>	It is easy to use and the application is easy to access.
NFR-5	<b>Availability</b>	The information whatever the user needed everything is available in this application.
NFR-6	<b>Scalability</b>	It is a open free source and support to all platform where millions of people can use this application.

## 5 PROJECT DESIGN

### 5.1 Data Flow Diagrams

## **5.2 Solution & Technical Architecture**

## **5.3 User Stories**

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	10 October 2022
Prepare Empathy Map	Prepare Empathy Map Canvasto capture the user Pains & Gains, Prepare list of problem statements	9 October 2022

Ideation Brain Storming	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	16 October 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	10 September 2022
Problem Solution Fit	Prepare problem solution fit document	16 October 2022
Solution Architecture	Prepare solution architecture document.	16 October 2022

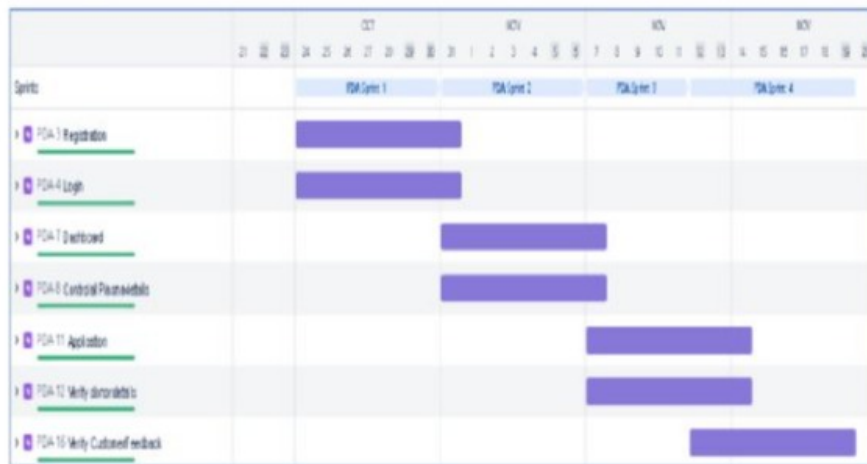


Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	16 October 2022
------------------	---	-----------------

Solution Requirement	Prepare the functional requirement document	16 October 2022
Data Flow Diagrams	Draw the dataflow diagrams and submit for review.	16 October 2022
Technology Architecture	Prepare the technology architecture diagram.	16 October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	26 October 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

## **6.2 Sprint Delivery Schedule**

### 6.3 Reports from JIRA



## 7. CODING & SOLUTIONING

### Login.html :

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login Form</title>

<link rel="stylesheet" href="{{ url_for('static',
filename='assets/css/loginstyle.css') }}">

</head>

<body>

<h1 style="color: white; margin: 6.5% 0 0 37.5%">Login...!</h1>
```

```
<form method="POST">
```

```
    <input type="email" placeholder="email" name="email" required>
```

```
    <input type="password" placeholder="password" name="password"
required>
```

```
    <input type="submit" value = "login">
```

```
    <span>Forgot <a href="">Password</a></span>
```

```
    <br>
```

```
    <br>
```

```
    <span>New User? <a href="{{ url_for('register') }}">Register</a></span>
```

```
</form>
```

```
</body>
```

```
</html>
```

```
<!-- {{ url_for('static', filename='assets/css/loginstyle.css') }} -->
```

**Register.html :**

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Login Form</title>
```

```
  <link rel="stylesheet" href="{{ url_for('static',  
filename='assets/css/loginstyle.css') }}">
```

```
</head>
```

```
<body>
```

```
  <h1 style="color: white; margin: 6.5% 0 0 38%">Register...</h1>
```

```
  <form method="POST">
```

```
    <input type="email" placeholder="email" name="email" required>
```

```
    <input type="password" placeholder="password" name="password"  
required>
```

<input type="submit" value = "Register">

<br>

<span>Already a user? <a href="{{url\_for('login')}}">Login</a></span>

</form>

</body>

</html>

## 8.TESTING

### 8.1 Test Cases

Fetch news	Functional	Home page	Verify user can get the news	News will be feed to the app	404 error	Fail	unsuccessful	App integration problem
Types of news available in the fetch news page	Functional	Fetch news	Types of news available	Weather, Sport, Economy.	Hover buttons will be shown.	Yes	successful	-

### Result:

Thus the News tracker application is developed and executed successfully.



## **10. ADVANTAGES & DISADVANTAGES**

### **Advantages:-**

- In this app news is already categorized.
- Easily accessible and portable.
- Better user experience.
- Apps help you convert visitors to loyal readers.
- Minute by minute updates of news.
- This app helps you to get local news updates instantly.
- To explore and discover trending news and topics.
- News feeds for you based on your interest.

### **Disadvantages:-**

- Some apps demand premium subscription from user.
- Occurrence of advertisement disturbs the user.
- Sometimes the news gives brief information.
- Prevalence of fake and uncertain news can confuse the user and lead to misconception.
- Fake news may mislead the readers.

## **11. CONCLUSION**

We explored the feasibility of recognising patterns of news reading interactions and evaluated three adaptive interface designs for different news reader types. We show that from their interaction log, a specific user can be recognised as one of three kinds. The reader types emerging from the online survey are well defined and distinct. The evaluation of the three variant interfaces suggests that different news reader types need different user interfaces. We have demonstrated a method for monitoring users' news reading behaviour and inferring news reader type from it. In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalisation of news apps.

## **12 . FUTURE SCOPE**

In the future we will further explore the design of adaptive interfaces, in order to be in a position to demonstrate a complete adaptive mobile news framework providing automatic personalisation of news apps.

## 13. APPENDIX

### Source Code: Tempates:

#### About.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Document</title>
```

```
<!-- CSS only -->
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
```

```
rel="stylesheet"
```

```
integrity="sha384-
```

```
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTR  
i"
```

crossorigin="anonymous">

<link rel="stylesheet" href=".static/assets/css/style.css">

<!-- scroll css -->

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swiper@8/swiper-  
bundle.min.css">

<!-- FONTS -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link  
href="https://fonts.googleapis.com/css2?family=Rubik+Glitch&display=swap"  
rel="stylesheet">

</head>

<body>

**Base.html:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<meta name="Description" content="Enter your description here"/>

<!-- Favicons -->

<link href="{{ url\_for('static', filename='/favicon.ico') }}" rel="icon">

<link rel="stylesheet"

href="https://fonts.googleapis.com/css?family=Overpass">

<link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel

min.css" integrity="sha512-

tS3S5qG0BlhnQROyJXvNjeEM4UpMXHrQfTGmbQ1gKmelCxlSEBUaxhRBj/

EFTzpbP4RVSRpEikbmdJobCvhE3g==" crossorigin="anonymous"

referrerpolicy="no-referrer" />

<link

href=

“<https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css>”

rel="stylesheet" integrity="sha384-

EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65Vohhpu  
uCOM

LASjC" crossorigin="anonymous"> <link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/font-

awesome/5.15.4/css/all.min.css">

<link rel="stylesheet" href="{{ url\_for('static',

filename='assets/css/style.css')

}}">

<!-- fonts -->

<!-- FONTS -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link

href="https://fonts.googleapis.com/css2?family=Rubik+Glitch&display=swap"

rel="stylesheet">

<title>Spot-on Times</title>

</head>

<body>

{% block navbar %}

<div class="container-fluid text-center" style="position: fixed; z-index:3;

margin-top: -6%; background-color: #16003B;">

<div class="row">

<div class="col">

<nav class="navbar">

<div class="container-fluid">

<a class="navbar-brand" href="#" style="font-family: 'Rubik Glitch',

cursive; color:white;font-size: 25px;">

```


BULLETIN

</a>

</div>

</nav>

</div>

<!-- search-bar -->

<div class="col" style="margin-top: 5px">

<nav class="navbar">

<div class="container-fluid">

<form class="d-flex" role="search">

<input class="form-control form-control-sm my-search-bar"

type="search" placeholder="Search"
```



```
        aria-label="Search">

        <button class="btn btn-outline-light"
type="submit">Search</button>

    </form>

</div>

</nav>

</div>


<div class="col" style="margin:auto">

    <nav class="nav justify-content-end">

        <a class="nav-link active" href="{{ url_for('headlines')
}}">Headlines</a>

        <a class="nav-link " href="{{ url_for('articles') }}">Articles</a>

        <a class="nav-link " href="{{ url_for('sources') }}">Sources</a>

        <a class="nav-link " href="#" style="background-color:white;
color:black;border-radius: 15%;">Login</a>

    </nav>
```

</div>

</div>

</div>

{% endblock %}

{% block content %}

{% endblock %}

{% block footer %}

<div class="container-fluid footer">

<p class="text-center footer-text">

© <span class="logo">Spot-On </span> Times

</p>

</div>

{% endblock %}

<script

src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.slim.min.js  
">

</script>

```
<script

src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.1/umd/popper.min.js">

</script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.6.0/js/bootstrap.min.js"></script>

</body>

</html>
```

### **Science.html:**

```
{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">

<div class="row">

{% for source, title, desc, author, img, p_date, url in sources %}
```

```
<div class="col-md-4 d-flex justify-content-center">
```

```
<div class="card mb-3 bg-dark text-white">
```

```

```

```
<div class="card-body">
```

```
<p class="card-text source">{{ source.name }}</p>
```

```
<a href="{{ url }}">
```

```
<h2 class="card-title"><b> {{ title }} </b></h2>
```

```
</a>
```

```
<p class="card-text">{{ desc }}</p>
```

```
<p class="card-text"><svg xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,
1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10
10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-
```

1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0

3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-

2.228z"></path></svg> Author: {{ author }}</p>

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg"

xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24

24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-

4.411 0-8 3.589-8 8s3.589 8 8 8-3.589 8-8 3.589-8-8-8zm0 14c-3.309 0-6-2.691-

6-6s2.691-6 6-6 6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9

2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published:

{{ p\_date }}</p>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

## Article.html

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<meta name="description" content="">
```

```
<meta name="author" content="Mark Otto, Jacob Thornton, and  
Bootstrapcontributors">
```

```
<meta name="generator" content="Hugo 0.84.0">
```

```
<title>Sign In</title>
```

```
<link rel="canonical"  
href="https://getbootstrap.com/docs/5.0/examples/sign-in/">
```

```
<link href="https://getbootstrap.com/docs/5.0/assets/css/docs.css"  
rel="stylesheet">
```

```
<!-- Bootstrapcore CSS -->
```

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.cs
s" rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZ
B6+fzT" crossorigin="anonymous">
```

```
<!-- Favicons -->
```

```
<link rel="apple-touch-icon" href="/docs/5.0/assets/img/favicons/apple-
touch-icon.png" sizes="180x180">
```

```
<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-
32x32.png" sizes="32x32" type="image/png">
```

```
<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-
16x16.png" sizes="16x16" type="image/png">
```

```
<link rel="manifest" href="/docs/5.0/assets/img/favicons/manifest.json">
```

```
<link rel="mask-icon" href="/docs/5.0/assets/img/favicons/safari-
pinned-tab.svg" color="#7952b3">
```

```
<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon.ico">
```

```
<meta name="theme-color" content="#7952b3">
```

```
<style>
```

```
.bd-placeholder-
```

```
img { font-size:
```

```
1.125rem; text-  
  
anchor: middle;  
  
-webkit-user-select:none;  
  
-moz-user-select:  
  
none;user-select:  
  
none;  
  
}  
  
@media (min-width: 768px) {  
  
.bd-placeholder-  
  
img-lg {font-size:  
  
3.5rem;  
  
}  
  
}  
  
</style>  
  
<!-- Customstyles for this template -->  
  
<link href="static/sheets/signin.css" rel="stylesheet">  
  
<link href="static/sheets/colors.css" rel="stylesheet">
```



</head>

<body class="text-center">

<nav class="navbar navbar-dark bg-dark fixed-top">

<div class="container-fluid">

<a class="navbar-brand" href="#">News Tracker</a>

<button class="navbar-toggler" type="button" data-  
bs-toggle="offcanvas" data-bs-  
target="#offcanvasDarkNavbar" aria-  
controls="offcanvasDarkNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<div class="offcanvas offcanvas-end text-bg-dark" tabindex="-1"  
id="offcanvasDarkNavbar" aria-labelledby="offcanvasDarkNavbarLabel">

<div class="offcanvas-header">

<h5 class="offcanvas-title"  
id="offcanvasDarkNavbarLabel">Profile</h5>

<button type="button" class="btn-close btn-close-white"  
data-bs-dismiss="offcanvas" aria-label="Close"></button>

</div>

```
<div class="offcanvas-body">
```

```
<ul class="navbar-nav justify-content-end flex-grow-1 pe-3">
```

```
<li class="nav-item">
```

```
<a class="nav-link active" aria-  
current="page" href="home.html">Home</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="base.html">Fetch News</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="about.html">About Us</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="signin.html">Sign In</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="signup.html">Sign Up</a>
```

```
</li>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
<main class="form-signin ">
```

```
<form action= "{{ url_for('getUser')}}" method="POST">
```

```

```

```
<h1 class="h3 mb-3 fw-normal white">Change your password</h1>
```

```
<div class="form-floating">
```

```
<input type="email" class="form-control" id="floatingInput"
name ="uname" placeholder="name@example.com">
```

```
<label for="floatingInput">Email address</label>
```

```
</div>
```

```
<br>
```

```
<button class="w-100 btn btn-lg btn-warning btn-outline-  
warning" type="submit"><a  
href="static/templates/changepass.html">Change  
Password</a></button><br><br>
```

```
</form>
```

```
</main>
```

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.m  
in.js" integrity="sha384-  
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPO  
Om Mi466C8" crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/@docsearch/js@3"></script>
```

```
<script  
src="https://cdn.jsdelivr.net/npm/@stackblitz/sdk@1/bundles/sdk.umd.j  
s">
```

```
</script>
```

```
<script src="/docs/5.2/assets/js/docs.min.js"></script>
```

```
<script>
```

```
document.querySelectorAll('.btn-edit').forEach(btn  
=> {btn.addEventListener('click', event => {
```

```
const htmlSnippet = event.target.closest('.bd-  
code-snippet').querySelector('.bd-  
example').innerHTML
```

```
const classes = Array.from(event.target.closest('.bd-  
code-snippet').querySelector('.bd-  
example').classList).join(' ')
```

```
const jsSnippet = event.target.closest('.bd-code-  
snippet').querySelector('.btn-edit').getAttribute('data-sb-js-snippet')
```

```
StackBlitzSDK.openBootstrapSnippet(htmlSnippet, jsSnippet, classes)
```

```
}}
```

```
}}
```

```
StackBlitzSDK.openBootstrapSnippet = (htmlSnippet, jsSnippet, classes)
```

```
=> {const markup = `<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.m
```

```
in.c ss" rel="stylesheet">
```

```
<link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css"
rel="stylesheet">
```

```
<title>Bootstrap Example</title>
```

```
<${'script'}
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></${'script'}>
```

```
</head>
```

```
<body class="p-3 m-0 border-0 ${classes}">
```

```
<!-- ExampleCode -->
```

```
${htmlSnippet.replace(/</gm, ' ')}>
```

```
<!-- End Example Code -->
```

```
</body>
```

```
</html>
```

const jsSnippetContent = jsSnippet ? `\\ NOTICE!!! Initially  
embedded in our docs this JavaScript\\n\\n file contains elements that can  
help you create reproducible\\n\\n use cases in StackBlitz for  
instance\\.\\n\\n In a real project please adapt this content to your



```

toggle=\u0022popover\u0022]\u0027)\n  \.forEach\(popover =\u003e
\n new bootstrap.Popover\(popover)\n \})\n\n \/\ \-\-\-\-\-\-\-\-\-\-\-\-
\-\-\-\-
\-\-\-\-\-\-\-\-\-\-\n \/\ Toasts\n \/\ \-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-
\-\-\-\-\n \/\ Used by \u0027Placement\u0027 example in docs or
StackBlitz\nconst toastPlacement =
document.getElementById(\u0027toastPlacement\u0027)\n if
(toastPlacement) {\n
document.getElementById(\u0027selectToastPlacement\u0027).addEven
tListener(\u0027change\u0027, function () {\n    if
(!toastPlacement.dataset.originalClass) {\n
toastPlacement.dataset.originalClass = toastPlacement.className\n
}\n\ntoastPlacement.className =
`${toastPlacement.dataset.originalClass}`
`${this.value}`\n    \})\n \})\n\n \/\ Instantiate all toasts in a docs page
only\ndocument.querySelectorAll(\u0027.bd-example .toast\u0027)\n
.forEach\(toastNode =\u003e {\n    const toast = new
bootstrap.Toast\(toastNode, {\n    autohide: false\n    \})\n\n
toast.show()\n    \})\n\n \/\ Instantiate all toasts in a docs page

```



only\nconsttoastTrigger =

```
document.getElementById(\u0027liveToastBtn\u0027)\n
```

```
const toastLiveExample =
```

```
document.getElementById(\u0027liveToast\u0027)\n if (\u0026toastTrigger\u0026)
```

```

\toastTrigger\addListener(\u0027click\u0027, \() =\u003e {\n

```

```
const toast = new bootstrap.Toast(toastLiveExample)\n\n
```

toast\show\(\)\n

**Alerts**

W Used in

\u0027Showlive toast\u0027 example in docs or StackBlitz\n const

alertPlaceholder =

```
document.getElementById(\u0027liveAlertPlaceholder\u0027)\n\nconst
```

alertTrigger =

```
document.getElementById(\u0027liveAlertBtn\u0027)\n\nconst
```

```
appendAlert = \ (message, type) => \u003e \{\n      const wrapper =
```

document.createElement('div')\n

```
wrapper\innerHTML = \[\\n
```

`\u003cdiv class=\u0022alert alert\-$\{type\} alert\-`

```

dismissible\alert\`,\n
\cdiv\${message}\c\div\`,\n
\cbbutton\button\ class=\btn\-close\
data-bs-dismiss=\alert\ aria-
label=\Close\c\button\,\n
\c\div\,\n
\].join(\)\n\n
alertPlaceholder.append(wrapper)\n\n if (alertTrigger)
{\n alertTrigger.addEventListener(click,\n)
=\` \nappendAlert(\`Nice, you triggered this alert
message!\,
\success\)\n \})\n \}\n\n
\ Checks \Radios\ \ \ \ \ \
\ Indeterminate checkbox example in docs and
StackBlitz\document.querySelector(\.bd-example-
indeterminate
[type=\checkbox\]\n \.forEach((checkbox =\`
{\nif (checkbox.id.includes(\`Indeterminate\`)) {\n

```



whatever\u0027)\n\n \u2218 Update the modal\u0027s content.\n

```
const modalTitle =
```

exampleModal\querySelector\(\u0027\modal\-title\u0027\)\n

```
const modalBodyInput =
```

```
exampleModal.querySelector(\u0027.modal-body
```

```
input\u0027)\n\n    modalTitle.textContent = `New message to`
```

$$\backslash \$\{\text{recipient}\} \backslash n \quad \text{modalBodyInput.value = recipient} \backslash n \quad \backslash \backslash \backslash n \quad \backslash \backslash \backslash n \quad \vee \vee \backslash -$$
[illegible]

\\ \u0027Offcanvas

components\u0027examplein docs only\n const myOffcanvas =

document.querySelectorAll(\u0027.bd-example-offcanvas

if \myOffcanvas\ \n

```
myOffcanvas.forEach(offcanvas => {\n
```

```
offcanvas\addEventListener(\u0027show.bs.offcanvas\u0027, event
```

=\u003e

```
\{n      event\preventDefault\(\)\n      \}, false\)\n      \}\)\n \}\n\}\)\n\}\n':
```

$$\text{nullconst project} = \{$$

```
files: {
```

'index.html': markup,

'index.js':

jsSnippetContent

},

title: 'Bootstrap Example',

description: `Official example from

`${window.location.href}``,template: jsSnippet ? 'javascript'

: 'html',

tags: ['bootstrap']

}

StackBlitzSDK.openProject(project, { openFile: 'index.html' })

}

</script>

</body>

</html>

## Home.html

```
{% extends 'base.html' %}
```

```
{% block content %}
```

```
<div class="container-fluid landing">
```

```
    <div class="row">
```

```
        {% for source, title, desc, author, img, p_date, url in sources %}
```

```
            <div class="col-md-4 d-flex justify-content-center">
```

```
                <div class="card mb-3 bg-dark text-white">
```

```
                    
```

```
                    <div class="card-body">
```

```
                        <p class="card-text source">{{ source.name }}</p>
```

```
                        <a href="{{ url }}">
```

```
                            <h2 class="card-title"><b> {{ title }} </b></h2>
```

```
                        </a>
```

```
                        <p class="card-text">{{ desc }}</p>
```

<p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8 8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6-2.691 6-6 6-2.691 6-6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p\_date }}</p>

</div>

</div>

</div>

```
{% endfor %}
```

```
</div>
```

```
</div>
```

```
{% endblock %}
```

## **Signin.html**

```
<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<meta name="description" content="">
```

```
<meta name="author" content="Mark Otto, Jacob Thornton, and  
Bootstrap contributors">
```

```
<meta name="generator" content="Hugo 0.84.0">
```

```
<title>Sign In</title>
```

```
<link rel="canonical"
```



```
href="https://getbootstrap.com/docs/5.0/examples/sign-in/">
```

```
<link href="https://getbootstrap.com/docs/5.0/assets/css/docs.css"
rel="stylesheet">
```

```
<!-- Bootstrapcore CSS -->
```

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.cs
s" rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZ
B6+fzT" crossorigin="anonymous">
```

```
<!-- Favicons-->
```

```
<link rel="apple-touch-icon" href="/docs/5.0/assets/img/favicons/apple-
touch-icon.png" sizes="180x180">
```

```
<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-
32x32.png" sizes="32x32" type="image/png">
```

```
<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-
16x16.png" sizes="16x16" type="image/png">
```

```
<link rel="manifest" href="/docs/5.0/assets/img/favicons/manifest.json">
```

```
<link rel="mask-icon" href="/docs/5.0/assets/img/favicons/safari-
pinned-tab.svg" color="#7952b3">
```

```
<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon.ico">
```

```
<meta name="theme-color" content="#7952b3">
```

```
<style>
```

```
.bd-placeholder-
```

```
img { font-size:
```

```
1.125rem; text-
```

```
anchor: middle;
```

```
-webkit-user-select:none;
```

```
-moz-user-select:
```

```
none;user-select:
```

```
none;
```

```
}
```

```
@media (min-width: 768px) {
```

```
.bd-placeholder-
```

```
img-lg {font-size:
```

```
3.5rem;
```

```
}
```

```
}
```

```
</style>
```

```
<!-- Customstyles for this template -->
```

```
<link href="static/sheets/signin.css" rel="stylesheet">
```

```
<link href="static/sheets/colors.css" rel="stylesheet">
```

```
</head>
```

```
<body class="text-center">
```

```
<nav class="navbar navbar-dark bg-dark fixed-top">
```

```
<div class="container-fluid">
```

```
<a class="navbar-brand" href="#">News Tracker</a>
```

```
<button class="navbar-toggler" type="button" data-  
bs- toggle="offcanvas" data-bs-  
target="#offcanvasDarkNavbar" aria-  
controls="offcanvasDarkNavbar">
```

```
<span class="navbar-toggler-icon"></span>
```

```
</button>
```

```
<div class="offcanvas offcanvas-end text-bg-dark" tabindex="-1"  
id="offcanvasDarkNavbar" aria-labelledby="offcanvasDarkNavbarLabel">
```

```
<div class="offcanvas-header">
```

```
<h5 class="offcanvas-title"
```

id="offcanvasDarkNavbarLabel">Profile</h5>

<button type="button" class="btn-close btn-close-white"  
data-bs-dismiss="offcanvas" aria-label="Close"></button>

</div>

<div class="offcanvas-body">

<ul class="navbar-nav justify-content-end flex-grow-1 pe-3">

<li class="nav-item">

<a class="nav-link active" aria-  
current="page" href="home.html">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="base.html">Fetch News</a>

</li>

<li class="nav-item">

<a class="nav-link" href="about.html">About Us</a>

</li>

<li class="nav-item">

```
<a class="nav-link" href="signin.html">Sign In</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="signup.html">Sign Up</a>
```

```
</li>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
<main class="form-signin ">
```

```
<form action = "{{url_for('getUser')}}"method="POST">
```

```

```

```
<h1 class="h3 mb-3 fw-normal white">Please sign in</h1>
```

```
<div class="form-floating">
```

```
<input type="email" class="form-control" id="floatingInput"  
name ="uname" placeholder="name@example.com">
```

<label for="floatingInput">Email address</label>

</div>

<br>

<div class="form-floating">

<input type="password" class="form-control" id="floatingPassword" name  
= "upwd" placeholder="Password">

<label for="floatingPassword">Password</label>

</div>

<div class="checkbox mb-3">

<label class="white">

<input type="checkbox" value="remember-me"> Rememberme

</label>

</div>


<button class="w-100 btn btn-lg bd-purple-500 btn-outline-  
warning" type="submit">Sign in</button><br><br>

<p class="white"><b>Forgot Password? <a  
href="forpass.html">Changehere</b></a></p>

**Don't have an account? [SignUp](#)**

**&copy; 2022–2023**

**Or**

**Sign in with **

```
<script>
```

```
document.querySelectorAll('.btn-edit').forEach(btn
```

```
=> {btn.addEventListener('click', event => {
```

```
    const htmlSnippet = event.target.closest('.bd-  
code-snippet').querySelector('.bd-  
example').innerHTML
```

```
    const classes = Array.from(event.target.closest('.bd-  
code-snippet').querySelector('.bd-  
example').classList).join(' ')
```

```
    const jsSnippet = event.target.closest('.bd-code-  
snippet').querySelector('.btn-edit').getAttribute('data-sb-js-snippet')
```

```
    StackBlitzSDK.openBootstrapSnippet(htmlSnippet, jsSnippet, classes)
```

```
  })
```

```
})
```

```
StackBlitzSDK.openBootstrapSnippet = (htmlSnippet, jsSnippet, classes)
```

```
=> {const markup = `<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```



```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
<link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
```

```
<title>Bootstrap Example</title>
```

```
<${'script'}
```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></${'script'}>
```

```
</head>
```

```
<body class="p-3 m-0 border-0 ${classes}">
```

```
<!-- ExampleCode -->
```

```
${htmlSnippet.replace(/</gm, ' ')}</div>
```

```
<!-- End Example Code -->
```

```
</body>
```

```
</html>
```

const jsSnippetContent = jsSnippet ? '\\ NOTICE!!! Initially embedded in our docs this JavaScript\\ file contains elements that can



```

\\ Used by Placement example in docs or
StackBlitz
const toastPlacement =
document.getElementById('toastPlacement')
if (toastPlacement) {
document.getElementById('selectToastPlacement').addEventListener(
  'change', function () {
    if (!toastPlacement.dataset.originalClass) {
      toastPlacement.dataset.originalClass = toastPlacement.className
    }
    toastPlacement.className =
      `${toastPlacement.dataset.originalClass}
      ${this.value}`
  })
  // Instantiate all toasts in a docs page
  only
  document.querySelectorAll('.bd-example .toast')
  .forEach((toastNode) => {
    const toast = new
    bootstrap.Toast(toastNode, {
      autohide: false
    })
    toast.show()
  })
  // Instantiate all toasts in a docs page
  only
  const toastTrigger =
  document.getElementById('liveToastBtn')
  const toastLiveExample =
  document.getElementById('liveToast')
  if (toastTrigger) {
    toastTrigger.addEventListener('click', () => {
      const toast = new bootstrap.Toast(toastLiveExample)
      toast.show()
    })
  }
  // Alerts
  // Used in

```

```

Showlive toast example in docs or StackBlitz
const
alertPlaceholder =
document.getElementById('liveAlertPlaceholder')
const
alertTrigger =
document.getElementById('liveAlertBtn')

appendAlert = (message, type) => {
  const wrapper =
document.createElement('div')

  wrapper.innerHTML = `
    <div class="alert alert-${type}" alert-
dismissible" role="alert">
      <div>${message}</div>
      <button type="button" class="btn-close"
data-bs-dismiss="alert" aria-
label="Close"></button>
    </div>
  `.join('')

  alertPlaceholder.append(wrapper)
  if (alertTrigger)
    alertTrigger.addEventListener('click', () => {
      appendAlert('Nice, you triggered this alert
message!',
        'success')
    })
  }

  Checks Radios
  Indeterminate checkbox example in docs and

```





```
    ${window.location.href}`,template: jsSnippet ? 'javascript'
    : 'html',
    tags: ['bootstrap']
  }

  StackBlitzSDK.openProject(project, { openFile: 'index.html' })
}

</script>

</body>

</html>
```

## **Business.html**

```
<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">
```

<meta name="description" content="">

<meta name="author" content="Mark Otto, Jacob Thornton, and  
Bootstrap contributors">

<meta name="generator" content="Hugo 0.84.0">

<title>Sign Up</title>

<link rel="stylesheet" type="text/css" href="sign in.css">

<link rel="canonical"  
href="https://getbootstrap.com/docs/5.0/examples/checkout/">

<link  
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.cs  
s" rel="stylesheet" integrity="sha384-  
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2  
QvZ6jIW3" crossorigin="anonymous">

<link rel="preconnect" href="https://fonts.gstatic.com">

<link  
href="https://fonts.googleapis.com/css2?family=Noto+Sans+JP:wght@300&dis  
play=swap" rel="stylesheet">

<link rel="preconnect" href="https://fonts.gstatic.com">

<link  
href="https://fonts.googleapis.com/css2?family=Nunito+Sans:ital@1&displ



ay= swap" rel="stylesheet">

<link rel="preconnect" href="https://fonts.gstatic.com">

<link

href="https://fonts.googleapis.com/css2?family=Oxygen:wght@300&display=swap" rel="stylesheet">

<link rel="preconnect" href="https://fonts.gstatic.com">

<link

href="https://fonts.googleapis.com/css2?family=Playfair+Display:wght@500&display=swap" rel="stylesheet">

<!-- Bootstrapcore CSS -->

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

<!-- Favicons -->

<link rel="apple-touch-icon" href="/docs/5.0/assets/img/favicons/apple-touch-icon.png" sizes="180x180">

<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-32x32.png" sizes="32x32" type="image/png">

<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-

16x16.png" sizes="16x16" type="image/png">

<link rel="manifest" href="/docs/5.0/assets/img/favicons/manifest.json">

<link rel="mask-icon" href="/docs/5.0/assets/img/favicons/safari-pinned-tab.svg" color="#7952b3">

<link rel="icon" href="/docs/5.0/assets/img/favicons/favicon.ico">

<meta name="theme-color" content="#7952b3">

<style>

table{

text-align: center;

border:3px solid

black; border-

collapse: collapse;

width: 100%;

height: 90px;

}

th{

border-collapse:

```
collapse; border: 2px  
  
solid black; font-  
  
family: 'Oxygen', serif;  
  
font-weight: 35px;  
  
font-size:  
  
25px;  
  
height:  
  
45px;  
  
width:  
  
60px;  
  
color:  
  
black;  
  
}
```

```
td{  
  
border-collapse:  
  
collapse;border: 2px  
  
solid white; width:  
  
60px;
```

```
height:
45px;

font-
size:20px;

font-weight: 67px;

font-family: 'Oxygen',
serif;background-color:
white; color: black;
}

.tabrow
{

background-color: #00ff80;

}

.tabdat{

background-color:
black;color: white;

font-weight: bold;

}
```

</style>

<link rel="stylesheet"  
href="colors.css">

<!-- Custom stylesfor this template-->

</head>

<body class="bd-yellow-600">

<nav class="navbar navbar-dark bg-dark fixed-top">

<div class="container-fluid">

<a class="navbar-brand" href="#">News Tracker</a>

<button class="navbar-toggler" type="button" data-bs-  
toggle="offcanvas" data-bs-target="#offcanvasDarkNavbar" aria-  
controls="offcanvasDarkNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<div class="offcanvas offcanvas-end text-bg-dark" tabindex="-1"  
id="offcanvasDarkNavbar" aria-labelledby="offcanvasDarkNavbarLabel">

<div class="offcanvas-header">

```
<h5 class="offcanvas-title"
id="offcanvasDarkNavbarLabel">Profile</h5>
```

```
<button type="button" class="btn-close btn-close-white"
data-bs-dismiss="offcanvas" aria-label="Close"></button>
```

```
</div>
```

```
<div class="offcanvas-body">
```

```
<ul class="navbar-nav justify-content-end flex-grow-1 pe-3">
```

```
<li class="nav-item">
```

```
<a class="nav-link active" aria-
current="page" href="home.html">Home</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="base.html">Fetch News</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
<a class="nav-link" href="about.html">About Us</a>
```

```
</li>
```

```
<li class="nav-item">
```

`<a class="nav-link" href="signin.html">Sign In</a>`

`</li>`

`<li class="nav-item">`

`<a class="nav-link" href="signup.html">Sign Up</a>`

`</li>`

`</div>`

`</div>`

`</div>`

`</nav>`

`<div class="container div2">`

`<main>`

`<div class="py-5 text-center">`

`<h2 class="text-white div2">Add User</h2>`

`</div>`

`<div class="col-md-7 col-lg-8">`

`<h4 class="mb-3 text-black">Account Details</h4>`

```
<form action= "{{ url_for('storedUser')}}" method="POST">

<div class="row g-3">

    <!--<div class="col-sm-6">-->

        <!--<label for="firstName" class="form-label text-
white">AccountNumber</label>-->

        <!--<input type="text" class="form-control"
id="acc_no" name="acc_no" placeholder="" value=""
required>-->

    <!--</div>-->

<div class="row g-3">

    <div class="col-sm-6">

        <label for="firstName" class="form-label text-
black">Firstname</label>

        <input type="text" class="form-control" id="firstName"
name = "fname" placeholder="" value="" required>

        <div class="invalid-
feedback">Valid first
name is required.

    </div>
```



</div>

<div class="col-sm-6">

<label for="lastName" class="form-label text-black">Lastname</label>

<input type="text" class="form-control" id="lastName" name="lname" placeholder="" value="" required>

<div class="invalid-feedback">Validlast name  
is required.

</div>

</div>

<div class="col-12">

<label for="floatingInput" class="form-label text-black">Email</label>

<input type="email" class="form-control" id="email" name="mail" placeholder="you@example.com">

</div>

<div class="col-12">

```
<label for="pass" class="form-label text-black">Password</label>
```

```
<input type="password" class="form-control" id="pass"
name ="npwd" placeholder="Enter a Valid Password" required>
```

```
</div>
```

```
<div class="col-12">
```

```
<label for="confpass" class="form-label text-
black">ConfirmPassword</label>
```

```
<input type="password" class="form-control"
id="confpass" name="cpwd" placeholder="Enter a Valid
Password" required>
```

```
</div>
```

```
<hr class="my-4">
```

```
<a href="#" alt="didnt connect"><button class="w-100 btn bd-
teal-700btn-outline-dark btn-lg" type="submit">Add
Account</button></a>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</main>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.m
in.js" integrity="sha384-
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPO
Om Mi466C8" crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/@docsearch/js@3"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/@stackblitz/sdk@1/bundles/sdk.umd.j
s">

</script>
```

```
<script src="/docs/5.2/assets/js/docs.min.js"></script>
```

```
<script>
```

```
document.querySelectorAll('.btn-edit').forEach(btn
```

```
=> {btn.addEventListener('click', event => {
```

```
const htmlSnippet = event.target.closest('.bd-
code-snippet').querySelector('.bd-
example').innerHTML
```

```
const classes = Array.from(event.target.closest('.bd-
code-snippet').querySelector('.bd-example').classList).join('
```

```
)
```

```
    const jsSnippet = event.target.closest('.bd-code-  
snippet').querySelector('.btn-edit').getAttribute('data-sb-js-  
snippet')
```

```
    StackBlitzSDK.openBootstrapSnippet(htmlSnippet, jsSnippet, classes)
```

```
  })
```

```
})
```

```
StackBlitzSDK.openBootstrapSnippet = (htmlSnippet, jsSnippet, classes)
```

```
=> {
```

```
    const markup = `<!doctype html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.m  
in.css" rel="stylesheet">
```

```
<link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css"  
rel="stylesheet">
```

&lt;title&gt;Bootstrap Example&lt;/title&gt;

 $\langle \mathcal{S} \rangle$ 

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"></script>
```

&lt;/head&gt;

<body class="p-3 m-0 border-0 \${classes}">

```
<!-- ExampleCode -->
```

```
{htmlSnippet.replace(/^/gm, '  ')}

```

&lt;!-- End Example Code --&gt;

&lt;/html&gt;`

[illegible]

Copyright 2011- 2022 The Bootstrap Authors  
Twitter, Inc.  
Licensed under the Creative Commons Attribution 3.0  
Unported License  
For details, see

<https://creativecommons.org/licenses/by/3.0/>

```
globalbootstrap: false  
use strict
```

```
// Tooltips  
// Instantiate all tooltips  
in docs or StackBlitz page  
document.querySelectorAll('[data-bs-toggle=tooltip']  
).forEach((tooltip) =>  
  new bootstrap.Tooltip(tooltip)
```

```
)  
// Popovers  
// Instantiate all popovers in a docs or StackBlitz page  
document.querySelectorAll('[data-bs-toggle=popover']  
).forEach((popover) =>  
  new bootstrap.Popover(popover))  
//
```

```
Toasts  
// Used by 'Placement' example in docs or  
StackBlitz  
const toastPlacement =  
document.getElementById('toastPlacement')  
if (toastPlacement) {  
  document.getElementById('selectToastPlacement').addEventListener('change', function () {  
    if (!toastPlacement.dataset.originalClass) {
```

```
toastPlacement\.dataset\.originalClass = toastPlacement\.className\n
}\n\ntoastPlacement\.className =
`$${toastPlacement\.dataset\.originalClass`}
`${this\.value}`\n    `})\n }\n\n \/\ Instantiate all toasts in a docs page
only\ndocument\.querySelectorAll\(`\u0027.bd-example .toast\u0027`\)\n
\.forEach\((toastNode =\u003e {\n    const toast = new
bootstrap\.Toast\((toastNode, {\n        autohide: false\n    })\)\n\n
toast\.show\(\)\n    `})\n }\n\n \/\ Instantiate all toasts in a docs page
only\nconst toastTrigger =
document\.getElementById\(`\u0027liveToastBtn\u0027`\)\n const
toastLiveExample =
document\.getElementById\(`\u0027liveToast\u0027`\)\n if `(toastTrigger)`
{\n toastTrigger\.addEventListener\(`\u0027click\u0027`, `()`=\u003e {\n
const toast = new bootstrap\.Toast\((toastLiveExample)\)\n\n
toast\.show\(\)\n
}\n }\n\n \/\ -|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|\n \/\ Alerts\n
\/\ -|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|\n \/\ Used in
\u0027Showlive toast\u0027 example in docs or StackBlitz\n const
alertPlaceholder =
document\.getElementById\(`\u0027liveAlertPlaceholder\u0027`\)\n const
alertTrigger =
document\.getElementById\(`\u0027liveAlertBtn\u0027`\)\n\n const
appendAlert = `(message, type)=\u003e {\n    const wrapper =
document\.createElement\(`\u0027div\u0027`\)\n
```

```

wrapper.innerHTML = `
<div class="alert alert-${type} alert-dismissible" role="alert">
  <div>${message}</div>
  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
</div>
`.join(`\n`);

alertPlaceholder.append(wrapper);
if (alertTrigger) {
  alertTrigger.addEventListener('click', () => {
    nappendAlert(`Nice, you triggered this alert message!`,
      'success');
  });
}

// Checks Radios
// Indeterminate checkbox example in docs and StackBlitz
document.querySelectorAll(`.bd-example-indeterminate
[type="checkbox"]`).forEach((checkbox) => {
  if (checkbox.id.includes('Indeterminate')) {
    checkbox.indeterminate = true;
  }
});

// Links
// Disable empty links in docs examples
only=document.querySelectorAll(`.bd-content

```



[illegible]

```

myOffcanvas.forEach(function(offcanvas) {
  offcanvas.addEventListener('show.bs.offcanvas', event
    => {
      event.preventDefault(),
      false
    })
})
})
):

const project = {
  files: {
    'index.html': markup,
    'index.js':
      jsSnippetContent
  },
  title: 'Bootstrap Example',
  description: `Official example from ${window.location.href}`,
  template: jsSnippet ? 'javascript' :
    'html',
  tags: ['bootstrap']
}

StackBlitzSDK.openProject(project, { openFile: 'index.html' })
}

</script>

```

</body>

</html>

## Entertainment.html

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">

<div class="row">

{% for source, title, desc, author, img, p\_date, url in sources %}

<div class="col-md-4 d-flex justify-content-center">

<div class="card mb-3 bg-dark text-white">



<div class="card-body">

<p class="card-text source">{{ source.name }}</p>

<a href="{{ url }}">

<h2 class="card-title"><b> {{ title }} </b></h2>

</a>

<p class="card-text">{{ desc }}</p>

<p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3 1.272-3 3s1.274 3 3 3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8 8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6-2.691 6-6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p\_date }}</p>

</div>

</div>

```
</div>
```

```
{% endfor %}
```

```
</div>
```

```
</div>
```

```
{% endblock %}
```

### **Headlines.html :**

```
{% extends 'base.html' %}
```

```
{% block content %}
```

```
<div class="container-fluid landing">
```

```
<div class="row">
```

```
{% for source, title, desc, author, img, p_date, url in sources %}
```

```
<div class="col-md-4 d-flex justify-content-center">
```

```
<div class="card mb-3 bg-dark text-white">
```

```

```

```
<div class="card-body">
```

```
<p class="card-text source">{{ source.name }}</p>
```

<a href="{{ url }}">

<h2 class="card-title"><b> {{ title }} </b></h2>

</a>

<p class="card-text">{{ desc }}</p>

<p class="card-text"><svg xmlns="http://www.w3.org/2000/svg"

width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10 10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88

4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg> Author: {{ author }}</p>

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg"

xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8 8-3.589 8-8 8-8zm0 14c-3.309 0-6-2.691-

6-6s2.691-6 6-6 6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published:

{{ p\_date }}</p>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

{% endblock %}

**Health.html :**

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">

<div class="row">

```

{% for source, title, desc, author, img, p_date, url in sources %}

<div class="col-md-4 d-flex justify-content-center">

    <div class="card mb-3 bg-dark text-white">

        <div class="card-body">

            <p class="card-text source">{{ source.name }}</p>

            <a href="{{ url }}">

                <h2 class="card-title"><b> {{ title }} </b></h2>

            </a>

            <p class="card-text">{{ desc }}</p>

            <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,
1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10
10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-
1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0
3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-
2.228z"></path></svg> Author: {{ author }}</p>

        </div>

    <div class="card-footer">

```



```

<p class="card-text published"><svg class="p-svg"
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24
24"

style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:;><path d="M12 5c-4.411
0-8 3.589-8 8s3.589 8 8 8-3.589 8-8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-
6s2.691-6 6-6 6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9
2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published:
{{ p_date }}</p>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

{% endblock %}

Science.html:

{% extends 'base.html' %}

```

{% block content %}

<div class="container-fluid landing">

<div class="row">

{% for source, title, desc, author, img, p\_date, url in sources %}

<div class="col-md-4 d-flex justify-content-center">

<div class="card mb-3 bg-dark text-white">



<div class="card-body">

<p class="card-text source">{{ source.name }}</p>

<a href="{{ url }}">

<h2 class="card-title"><b> {{ title }} </b></h2>

</a>

<p class="card-text">{{ desc }}</p>

<p class="card-text"><svg xmlns="http://www.w3.org/2000/svg"

width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,

1);transform: ;msFilter:;"><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10

10-4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-  
1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0  
3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-  
2.228z"></path></svg> Author: {{ author }}

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg"  
xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24  
24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-  
4.411 0-8 3.589-8 8s3.589 8 8 8-3.589 8-8 3.589-8-8-8zm0 14c-3.309 0-6-2.691-  
6-6s2.691-6 6-6 6-2.691 6-6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9  
2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published:  
{{ p\_date }}

</div>

</div>

</div>

{% endfor %}

</div>

</div>

{% endblock %}

## Sources.html :

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">

<div class="row">

{% for source in newsSource %}

<div class="col-md-3 d-flex justify-content-center source-body">

<div class="card bg-dark">

<div class="card-body">

<a href="{{ source.url }}" target="\_blank">

<h5 class="card-title"><svg

xmlns="http://www.w3.org/2000/svg" width="36" height="36" viewBox="0 0 24

24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12

2C6.486 2 2 6.486 2 12s4.486 10 10 10-4.486 10-10S17.514 2 12 2zM4 12c0-

.899.156-1.762.431-2.569L6 11l2 2v2l2 2 1 1v1.931C7.061 19.436 4 16.072 4  
12zm14.33 4.873C17.677 16.347 16.687 16 16 16v-

1a2 2 0 0 0-2-2h-4v-3a2 2 0 0 0 2-2V7h1a2 2 0 0 0 2-2v-.411C17.928 5.778 20  
8.65 20 12a7.947 7.947 0 0 1-1.67 4.873z"></path></svg> {{ source.name  
}}</h5>

<p class="card-text">{{ source.description }}</p>

</a>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

{% endblock %}

**Sport.html:**

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">

<div class="row">

{% for source, title, desc, author, img, p\_date, url in sources %}

<div class="col-md-4 d-flex justify-content-center">

<div class="card mb-3 bg-dark text-white">



<div class="card-body"

<p class="card-text source">{{ source.name }}</p>

<a href="{{ url }}">

<h2 class="card-title"><b> {{ title }} </b></h2>

</a>

<p class="card-text">{{ desc }}</p>

<p class="card-text"><svg xmlns="http://www.w3.org/2000/svg"

width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,

1);transform: ;msFilter:; "><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10 10-

4.579 10-10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-

3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88

4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg>

Author: {{ author }}</p>

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg"

xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24"

style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-4.411 0-8

3.589-8 8s3.589 8 8 8-3.589 8-8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6

6-6 6 2.691 6 6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9

2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{

p\_date }}</p>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

```
{% endblock %}
```

## **Tech.html :**

```
{% extends 'base.html' %}
```

```
{% block content %}
```

```
<div class="container-fluid landing">
```

```
    <div class="row">
```

```
        {% for source, title, desc, author, img, p_date, url in sources %}
```

```
        <div class="col-md-4 d-flex justify-content-center">
```

```
            <div class="card mb-3 bg-dark text-white">
```

```
                
```

```
                <div class="card-body">
```

```
                    <p class="card-text source">{{ source.name }}</p>
```

```
                    <a href="{{ url }}">
```

```
                        <h2 class="card-title"><b> {{ title }} </b></h2>
```

```
                    </a>
```

```
                    <p class="card-text">{{ desc }}</p>
```

```
                    <p class="card-text"><svg xmlns="http://www.w3.org/2000/svg"
```



width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10-4.579 10-10 10S17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg>

Author: {{ author }}</p>

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg"

xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8-3.589 8-8 8-3.589-8-8-8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6-2.691 6-6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p\_date }}</p>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

{% endblock %}

**Today's news.html :**

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">

<div class="row">

{% for source, title, desc, author, img, p\_date, url in today's\_news %}

<div class="col-md-4 d-flex justify-content-center">

<div class="card mb-3 bg-dark text-white">



<div class="card-body">

<p class="card-text source">{{ source.name }}</p>

<a href="{{ url }}">

<h2 class="card-title"><b> {{ title }} </b></h2>

</a>

<p class="card-text">{{ desc }}</p>

<p class="card-text"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 2C6.579 2 2 6.579 2 12s4.579 10 10 10-4.579 10-10s17.421 2 12 2zm0 5c1.727 0 3 1.272 3 3s-1.273 3-3 3c-1.726 0-3-1.272-3-3s1.274-3 3-3zm-5.106 9.772c.897-1.32 2.393-2.2 4.106-2.2h2c1.714 0 3.209.88 4.106 2.2C15.828 18.14 14.015 19 12 19s-3.828-.86-5.106-2.228z"></path></svg>  
Author: {{ author }}</p>

</div>

<div class="card-footer">

<p class="card-text published"><svg class="p-svg" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);transform: ;msFilter:; "><path d="M12 5c-4.411 0-8 3.589-8 8s3.589 8 8 8-3.589 8-8 8zm0 14c-3.309 0-6-2.691-6-6s2.691-6 6-6 6-2.691 6-6-2.691 6-6 6z"></path><path d="M11 9h2v5h-2zM9 2h6v2H9zm10.293 5.707-2-2 1.414-1.414 2 2z"></path></svg> Date Published: {{ p\_date }}</p>

</div>

</div>

</div>

{% endfor %}

</div>

</div>

{% endblock %}

### **Init.py :**

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
app.config['SECRET_KEY'] = 'thisisasecretkey'
```

```
from app import views
```

## Config.py :

```
NEWS_BASE_URL_SOURCES = 'https://newsapi.org/v2/top-
headlines/sources?apiKey={}'

NEWS_BASE_EVERYTHING_URL =
'https://newsapi.org/v2/everything?domains={}&apiKey={}'

NEWS_BASE_HEADLINES_URL = 'https://newsapi.org/v2/top-
headlines?country=in&apiKey={}'

NEWS_BASE_SOURCE = 'https://newsapi.org/v2/top-
headlines?sources={}&apiKey={}'

API_KEY = "7d74e79ab06d438a89e5eb6f7a8deb40"

class ProdConfig(Config):

    pass

class DevConfig(Config):

    DEBUG = True

config_options= {
```

```
'development': DevConfig,  
  
'production': ProdConfig  
  
}
```

### **Model.py:**

```
class Sources:
```

```
    def __init__(self, name, description, url):
```

```
        self.name=name,
```

```
        self.description=description
```

```
        self.url=url
```

```
class Articles:
```

```
    """Define article model"""
```

```
    def __init__(self, source, author, title, description, url, urlToImage, publishedAt):
```

```
        self.source = source
```

```
        self.author = author
```

```
        self.title = title
```

```
        self.description = description
```

```
self.url = url
```

```
self.urlToImage = urlToImage
```

```
self.publishedAt = publishedAt
```

### **Request.py :**

```
from app.models import Articles
```

```
from app.models import Sources
```

```
from newsapi import NewsApiClient
```

```
from app.config import Config
```

```
import urllib.request,json
```

```
api_key=None
```

```
base_url=None
```

```
base_url_for_everything=None
```

```
base_url_top_headlines=None
```

```
base_source_list=None
```

```
all_types_articles = []
```

```
def news_sorter_3(all_articles):
```

```
source = []
```

```
title = []
```

```
desc = []
```

```
author = []
```

```
img = []
```

```
p_date = []
```

```
url = []
```

```
for i in range(3):
```

```
    article = all_articles[i]
```

```
    source.append(article['source'])
```

```
    title.append(article['title'])
```

```
    desc.append(article['description'])
```

```
    author.append(article['author'])
```

```
    img.append(article['urlToImage'])
```

```
    p_date.append(article['publishedAt'])
```

```
    url.append(article['url'])
```



```
article_object = Articles(source, title, desc, author, img, p_date, url)
```

```
all_types_articles.append(article_object)
```

```
contents = zip(source, title, desc, author, img, p_date, url)
```

```
return contents
```

```
def news_sorter_5(all_articles):
```

```
    source = []
```

```
    title = []
```

```
    desc = []
```

```
    author = []
```

```
    img = []
```

```
    p_date = []
```

```
    url = []
```

```
    for i in range(20):
```

```
article = all_articles[i]

source.append(article['source'])

title.append(article['title'])

desc.append(article['description'])

author.append(article['author'])

img.append(article['urlToImage'])

p_date.append(article['publishedAt'])

url.append(article['url'])
```

```
article_object = Articles(source, title, desc, author, img, p_date, url)

all_types_articles.append(article_object)

contents = zip(source, title, desc, author, img, p_date, url)
```

```
return contents
```

```
def news_sorter(all_articles):
```

```
    source = []
```

```
    title = []
```

```
desc = []
```

```
author = []
```

```
img = []
```

```
p_date = []
```

```
url = []
```

```
for i in range(len(all_articles)):
```

```
    article = all_articles[i]
```

```
    source.append(article['source'])
```

```
    title.append(article['title'])
```

```
    desc.append(article['description'])
```

```
    author.append(article['author'])
```

```
    img.append(article['urlToImage'])
```

```
    p_date.append(article['publishedAt'])
```

```
    url.append(article['url'])
```

```
    article_object = Articles(source, title, desc, author, img, p_date, url)
```

```
    all_types_articles.append(article_object)
```

```
contents = zip(source, title, desc, author, img, p_date, url)
```

```
return contents
```

```
def publishedArticlesReduced():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    news= []
```

```
    #todays news
```

```
    get_articles = newsapi.get_everything(sources= 'cnbc, NDTV News, Google News,  
Crictracker.com, The Indian Express, Hindustan Times, the-verge, gizmodo, the-next-  
web, techradar, recode, ars-technica')
```

```
    all_articles = get_articles['articles']
```

```
    output = news_sorter_3(all_articles)
```

```
    news.append(output)
```

```
    #top headlines
```

```
get_articles = newsapi.get_top_headlines(sources= 'NDTV News, Google News,  
Crictracker.com, The Indian Express, Hindustan Times, cnbc, techcrunch, the-verge,  
gizmodo, the-next-web, techradar, recode, ars-technica')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter_5(all_articles)
```

```
news.append(output)
```

```
#business articles
```

```
get_articles = newsapi.get_top_headlines(category='business')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter_5(all_articles)
```

```
news.append(output)
```

```
#tech articles
```

```
get_articles = newsapi.get_top_headlines(category='technology')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter_5(all_articles)
```

```
news.append(output)
```

```
#entertainment_articles
```

```
get_articles = newsapi.get_top_headlines(category='entertainment')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter_5(all_articles)
```

```
news.append(output)
```

```
#science_articles
```

```
science_articles = newsapi.get_top_headlines(category='science')
```

```
all_articles = science_articles['articles']
```

```
output = news_sorter_5(all_articles)
```

```
news.append(output)
```

```
#sport articles
```

```
sport_articles = newsapi.get_top_headlines(category='sports')
```

```
all_articles = sport_articles['articles']
```

```
output = news_sorter_5(all_articles)
```

```
news.append(output)
```

```
#health articles
```

```
get_articles = newsapi.get_top_headlines(category='health')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter_5(all_articles)
```

```
news.append(output)
```

```
return news
```

```
#
```

```
def publishedArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    news= []
```

```
    #todays news
```

```
    get_articles = newsapi.get_everything(sources= 'cnn, reuters, cnbc, the-verge,  
gizmodo, the-next-web, techradar, recode, ars-technica')
```

```
    all_articles = get_articles['articles']
```

```
    output = news_sorter(all_articles)
```

```
    news.append(output)
```

```
    #top headlines
```

```
    get_articles = newsapi.get_top_headlines(sources= 'cnn, reuters, cnbc, techcrunch,
```



the-verge, gizmodo, the-next-web, techradar, recode, ars-technica')

```
all_articles = get_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#business articles
```

```
get_articles = newsapi.get_top_headlines(category='business')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#tech articles
```

```
get_articles = newsapi.get_top_headlines(category='technology')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#entertainment_articles
```

```
get_articles = newsapi.get_top_headlines(category='entertainment')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#science_articles
```

```
science_articles = newsapi.get_top_headlines(category='science')
```

```
all_articles = science_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#sport articles
```

```
sport_articles = newsapi.get_top_headlines(category='sports')
```

```
all_articles = sport_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
#health articles
```

```
get_articles = newsapi.get_top_headlines(category='health')
```

```
all_articles = get_articles['articles']
```

```
output = news_sorter(all_articles)
```

```
news.append(output)
```

```
return news
```

```
def topHeadlines():
```

```
newsapi = NewsApiClient(api_key= Config.API_KEY)

top_headlines = newsapi.get_top_headlines(sources= 'cnn, reuters, cnbc,
techcrunch, the-verge, gizmodo, the-next-web, techradar, recode, ars-technica')

all_headlines = top_headlines['articles']


return news_sorter(all_headlines)


def randomArticles():

    newsapi = NewsApiClient(api_key= Config.API_KEY)

    random_articles = newsapi.get_everything(sources= 'the-verge, gizmodo, the-next-
web, recode, ars-technica')

    all_articles = random_articles['articles']


    return news_sorter(all_articles)


def businessArticles():

    newsapi = NewsApiClient(api_key= Config.API_KEY)

    business_articles = newsapi.get_top_headlines(category='business')

    all_articles = business_articles['articles']
```

```
return news_sorter(all_articles)
```

```
def techArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    tech_articles = newsapi.get_top_headlines(category='technology')
```

```
    all_articles = tech_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def entArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    ent_articles = newsapi.get_top_headlines(category='entertainment')
```

```
    all_articles = ent_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def scienceArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
science_articles = newsapi.get_top_headlines(category='science')
```

```
all_articles = science_articles['articles']
```

```
return news_sorter(all_articles)
```

```
def sportArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    sport_articles = newsapi.get_top_headlines(category='sports')
```

```
    all_articles = sport_articles['articles']
```

```
    return news_sorter(all_articles)
```

```
def healthArticles():
```

```
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    health_articles = newsapi.get_top_headlines(category='health')
```

```
    all_articles = health_articles['articles']
```

```
    return news_sorter(all_articles)
```

```

def get_news_source():
    """
    Function that gets the json response to our url request
    """

    get_news_source_url = 'https://newsapi.org/v2/sources?apiKey=' +
    Config.API_KEY

    with urllib.request.urlopen(get_news_source_url) as url:

        get_news_source_data = url.read()

        get_news_source_response = json.loads(get_news_source_data)

        news_source_results = None

        if get_news_source_response['sources']:

            news_source_results_list = get_news_source_response['sources']

            news_source_results = process_sources(news_source_results_list)

    return news_source_results

def process_sources(source_list):
    """

```

function that process the news articles and transform them to a list of objects

'''

```
news_source_result = []
```

```
for news_source_item in source_list:
```

```
    name = news_source_item.get('name')
```

```
    description = news_source_item.get('description')
```

```
    url = news_source_item.get('url')
```

```
    if name:
```

```
        news_source_object = Sources(name, description,url)
```

```
        news_source_result.append(news_source_object)
```

```
return news_source_result
```

### **Views.py:**

```
from app import app
```

```
from flask import session, flash, redirect, render_template, request, url_for
```

```
import sys
```

```
import ibm_db
```

```
from .request import businessArticles, entArticles, get_news_source,
```



healthArticles,publishedArticlesReduced, publishedArticles, randomArticles,  
scienceArticles, sportArticles, techArticles, topHeadlines

```
conn = ibm_db.connect(  
    'DATABASE=bludb;  
    'HOSTNAME=b1bc1829-6f45-4cd4-bef4-  
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;  
    'PORT=32304;  
    'SECURITY=SSL;  
    'SSLServerCertificate=DigiCertGlobalRootCA.crt;  
    'UID=pbd63698;  
    'PWD=M58PuW0KLPnAYIXM;', ", "  
)
```

global account

```
@app.route('/', methods=['GET', 'POST'])
```

```
def login():

    if request.method == 'POST':

        # getting user data

        email = request.form.get('email')

        password = request.form.get('password')

        sql_check_query = "SELECT * FROM user WHERE email = ?"

        stmt = ibm_db.prepare(conn, sql_check_query)

        ibm_db.bind_param(stmt, 1, email)

        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)

        print(account)

        if account:

            # email id exists

            # checking if the password is correct

            if not account['PASSWORD'] == password:

                flash('Invalid password', category='error')

            else:

                # user entered the correct password
```

```
        # redirecting the user to the dashboard

        session['user_id'] = account['EMAIL']

        return redirect(url_for('home'))

    else:

        # email id does not exist in the database

        flash('Email invalid... Try Again', category='error')

    return render_template('auth/login.html')

return render_template('auth/login.html')

# return render_template('login.html')

@app.route('/register', methods=['GET', 'POST'])
def register():

    if request.method == 'POST':

        # getting user data

        email = request.form.get('email')

        password = request.form.get('password')
```

```
# checking: user already exists or not

sql_check_query = "SELECT * FROM user WHERE email = ?"

stmt = ibm_db.prepare(conn, sql_check_query)

ibm_db.bind_param(stmt, 1, email)

ibm_db.execute(stmt)


account = ibm_db.fetch_assoc(stmt)

# email id does not exist in the database

if not account:

    # inserting the data into the database

    sql_insert_query = "INSERT INTO user VALUES (?, ?)"

    stmt = ibm_db.prepare(conn, sql_insert_query)

    ibm_db.bind_param(stmt, 1, email)

    ibm_db.bind_param(stmt, 2, password)

    ibm_db.execute(stmt)


# user data inserted into the database

# redirecting to login page

flash('User created successfully! Please Login', category='success')
```

```
return redirect('/')
```

```
else:
```

```
    flash('Email id already exists! Try another one', category='error')
```

```
return render_template('auth/register.html')
```

```
return render_template('auth/register.html')
```

```
@app.route('/home')
```

```
def home():
```

```
    articles = publishedArticlesReduced()
```

```
    todays_news = articles[0]
```

```
    top_headlines = articles[1]
```

```
    business_articles = articles[2]
```

```
    tech_articles = articles[3]
```

```
    entertainment_articles = articles[4]
```

```
    science_articles = articles[5]
```

```
    sport_articles = articles[6]
```

```
health_articles = articles[7]
```

```
print(todays_news, file=sys.stderr)
```

```
return render_template('home.html',
```

```
    todays_news = todays_news,
```

```
    top_headlines = top_headlines,
```

```
    business_articles = business_articles,
```

```
    tech_articles = tech_articles,
```

```
    entertainment_articles = entertainment_articles,
```

```
    science_articles = science_articles,
```

```
    sport_articles = sport_articles,
```

```
    health_articles = health_articles,
```

```
    business_BOTTOM = business_articles,
```

```
    random_articles=todays_news)
```

```
@app.route('/todaysnews')
```

```
def todaysnews():
```

```
    articles = publishedArticles()
```

```
    todays_news = articles[0]
```

```
return render_template('todaysnews.html',todays_news = todays_news)
```

```
@app.route('/headlines')
```

```
def headlines():
```

```
    headlines = topHeadlines()
```

```
    return render_template('headlines.html', headlines = headlines)
```

```
@app.route('/articles')
```

```
def articles():
```

```
    random = randomArticles()
```

```
    return render_template('articles.html', random = random)
```

```
@app.route('/sources')
```

```
def sources():
```

```
    newsSource = get_news_source()
```

```
return render_template('sources.html', newsSource = newsSource)
```

```
@app.route('/category/business')
```

```
def business():
```

```
    sources = businessArticles()
```

```
    return render_template('business.html', sources = sources)
```

```
@app.route('/category/tech')
```

```
def tech():
```

```
    sources = techArticles()
```

```
    return render_template('tech.html', sources = sources)
```

```
@app.route('/category/entertainment')
```

```
def entertainment():
```

```
    sources = entArticles()
```

```
    return render_template('entertainment.html', sources = sources)
```



```
@app.route('/category/science')
```

```
def science():
```

```
    sources = scienceArticles()
```

```
    return render_template('science.html', sources = sources)
```

```
@app.route('/category/sports')
```

```
def sports():
```

```
    sources = sportArticles()
```

```
    return render_template('sport.html', sources = sources)
```

```
@app.route('/category/health')
```

```
def health():
```

```
    sources = healthArticles()
```

```
    return render_template('health.html', sources = sources)
```

## **GitHub**

<https://github.com/IBM-EPBL/IBM-Project-39895-1660561693>

## **Project demo link :**

<https://drive.google.com/file/d/1sBEmqVedX4spJuHIX0aCF-vjNLZKSYxl/view?usp=drivesdk>