

**HX8001 PROFESSIONAL READINESS FOR INNOVATION,
EMPLOYABILITY AND ENTREPRENEURSHIP.**

Submitted by

TEAM ID:PNT2022TMID40132

DURAI RAJ S - 512219104006

SHIVARAM M - 512219104017

SIVANESAN R - 512219104018

SATHISH KUMAR R - 512219104016

Pursuing final year of the degree

BACHELOR OF ENGINEERING

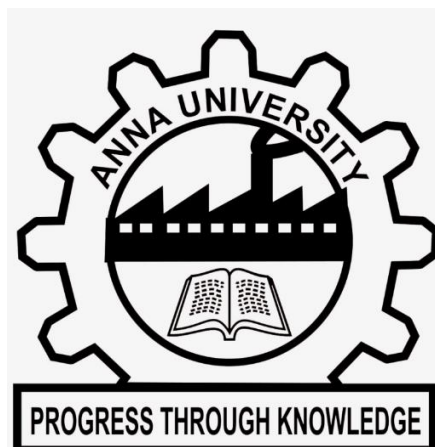
In

**INFORMATION TECHNOLOGY & COMPUTER SCIENCE
AND ENGINEERING**

SKP ENGINEERING COLLEGE,

THIRUVANNAMALAI – 606611

NOV 2022



ANNA UNIVERSITY: CHENNAI 600 025

ACKNOWLEDGEMENT

NEWS TRACKER APPLICATION

IN CLOUD COMPUTING

I am very thankful to the management of SKP Engineering College, Tiruvannamalai for giving us this wonderful opportunity to study in this college and utilize all the facilities to the fullest.

My sincere thanks to lord blessing on us for the successful completion of My project. I thank proudly **Mr. K. KARUNANITHI, B.E., MBA**, Chairman of my college for providing the facility to do the mini project. I am grateful for my Principal **Dr.S. BASKARAN, M.E, Ph.D.**, for his constant support and encouragement to our project.

I express my sincere thanks to **Dr.N.PURUSHOTHAMAN,M.E, Ph.D.**, Head of the Department of Computer science and Supervisor **L. SANKARAN, M.E** for us for guidance, timely suggestions and word motivation.

It is a Great pleasure to express my gratitude and thanks towards my project Guide **L. SANKARAN, M.E** for his uninterrupted suggestions and words of improvements regarding this mini project, which played a major role in guiding me in my track.

We also extend our thanks to our Class Advisor and every member of the faculty who provided valuable academic guidance, and their co-operation to do this successful project. Finally, we would like to thank our parents, relatives and friends for their encouragement and enthusiastic co-operation.

TABLE OF FIGURES

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

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

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX Source Code GitHub & Project Demo Link

1 INTRODUCTION

1.1 PROJECT OVERVIEW

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.

1.2 PURPOSE

A news application is a big interactive database that tells a news story. Think of it like you would any other piece of journalism. It just uses software instead of words and pictures.

By showing each reader data that is specific to them, a news app can help each reader understand a story in a way that's personally meaningful to them. It can help a reader understand their personal connection to a broad national phenomenon, and help them attach what they know to what they don't know, and thereby encourage a deep understanding of abstract concepts.

A news app should tell a story, and just like any good news story, it needs a headline, a byline, a lead, and a nut graph. Some of these concepts can be hard to distinguish in a piece of interactive software, but they're there if you look closely.

Also, a news app should be generative — meaning it should generate more stories and more reporting. ProPublica's best apps have been used as the basis for local stories.

2. LITERATURE SURVEY

2.1 Existing System

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 References

- Digital Journalism by : Oscar Westland (2013) MOBILE NEWS, , Digital J
- Exploring mobile news reading interactions for news app personalisation by: Mobile HCI '15, August 25 - 28, 2015, Copenhagen, Denmark © 2015 ACM

2.3 Problem Statement Definition

- 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.
- The user wants to read news only about particular topics so that he can be informed about his interest
- The user wants to get informed from only certified news outlets .
- The user needs a way to search about the news on topics he wants to.
- The user wants to know the news about his surrounding using GPS location.

3. IDEATION & PROPOSED SOLUTION


3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 IDEATION AND BRAINSTROMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

🕒 10 minutes to prepare
🕒 1 hour to collaborate
👤 2-8 people recommended

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

🕒 10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the facilitation Superpowers to run a happy and productive session.

[Open article](#) →

1 Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM STATEMENT
User needs a way to get a interested marketing in a quick peek of the day. So that we give relevant news where the user more interested on that day and save there time.

Key rules of brainstorming
To run an smooth and productive session

🗣️ Stay in topic.	💡 Encourage wild ideas.
🚫 Defer judgment.	👂 Listen to others.
🗣️ Go for volume.	🖼️ If possible, be visual.

3

Brainstorm

Write down any ideas that come to mind that address your problem statement.

20 minutes

Round 1

Round 2

Round 3

Round 4

4

Organize

Take your divergent ideas and cluster them together in related boxes or groups. Write the 3-5 most important ideas in a central box. The cluster is bigger than the ideas within it, and you will use it to create your solution.

20 minutes

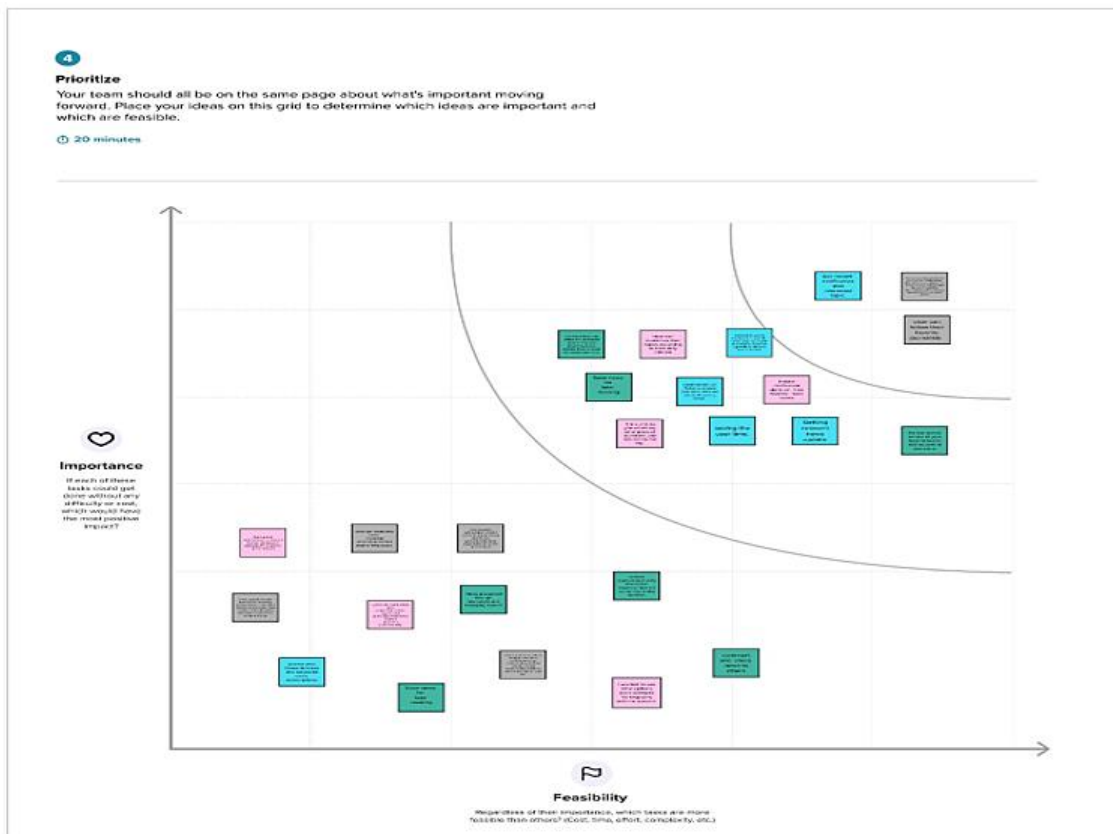
Use the following categories to organize your ideas:

User Identification

User Identification

User Interface and Usability

Technology

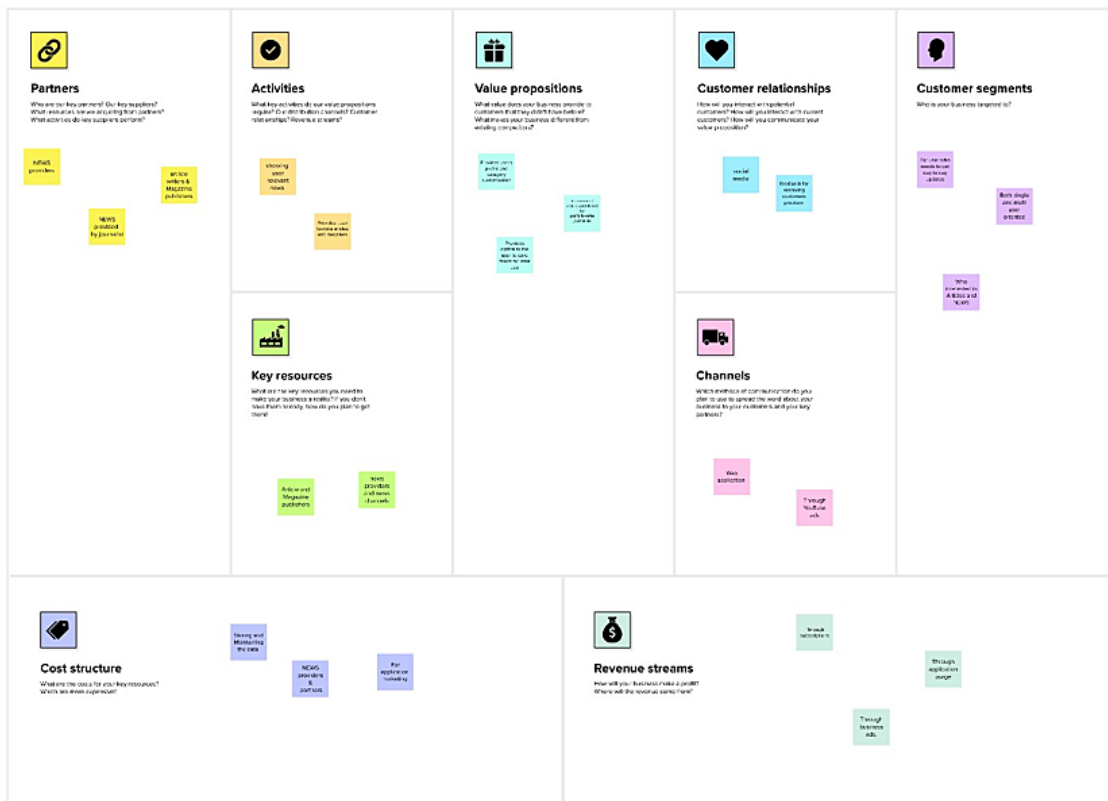


3.3 PROPOSED SOLUTION

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	User needs a way to get a interested marketing in a quick peek of the day. So that we give relevant news where the user more interested on that day and save their time.
2.	Idea / Solution description	We provide relevant news where the user more interested on that day.
3.	Novelty / Uniqueness	Saving and downloading the interested articles.
4.	Social Impact / Customer Satisfaction	When news feed are customized/personalized to the user, the time spend to learn about the day will be minimized and will be completely productive.
5.	Business Model (Revenue Model)	Shown below.
6.	Scalability of the Solution	<p>The first step is to predict user interests and collect relevant data for user personalization.</p> <p>The system provides filter option and sorting the user relevant news and saves the time.</p>

Business Model:



The work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

3.4 PROBLEM SOLUTION FIT

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) <small>Who is your customer? i.e. working parents of 0.5 yo. kids</small> Customers who wants the glimpse of the daily NEWS.	6. CUSTOMER CONSTRAINTS <small>What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.</small> There is no time to read the NEWS in current busy world.	5. AVAILABLE SOLUTIONS <small>Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking</small> Provide user interesting glimpse to save the users times.	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS <small>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.</small> Collecting the data on daily NEWS and providing the glimpse. Customize the NEWS base on the customer interest.	9. PROBLEM ROOT CAUSE <small>What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.</small> There is no time to read the big paragraphs.	7. BEHAVIOUR <small>What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)</small> Customer getting irritated after seeing the long paragraph.	
Focus on J&P, tap into BE, understand RC	3. TRIGGERS <small>What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.</small> Send a notification to the customer where customer interested topics.	10. YOUR SOLUTION <small>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.</small> we provide the NEWS where the customer interested topics. And avoid the long paragraphs and give the glimpse.	8. CHANNELS of BEHAVIOUR 8.1 ONLINE <small>What kind of actions do customers take online? Extract online channels from #7</small> 8.2 OFFLINE <small>What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</small> Anytime, <u>anywhere</u> access to the latest and updated news through the online channels. When the user in online they download the article. Afterwards they see <u>the download</u> articles in offline.	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER <small>How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.</small> They feel irritated will see the long paragraphs.			
	After facing the <u>problem</u> they feel don't want to use the platform.			

4. REQUIREMENT ANALYSIS

Functional Requirements:

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 Facebook Registration through Gmail Registration through Mobile Number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Confirmation via Mobile OTP
FR-3	Favourites	Search and follow favourite feeds

FR-4	NEWS Feed	The news feed is generated using the posts from other entities in the system that the user followed or the user might be interested in.
FR-5	Posts	The posts might have text, image and video.

Non-functional Requirements:

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

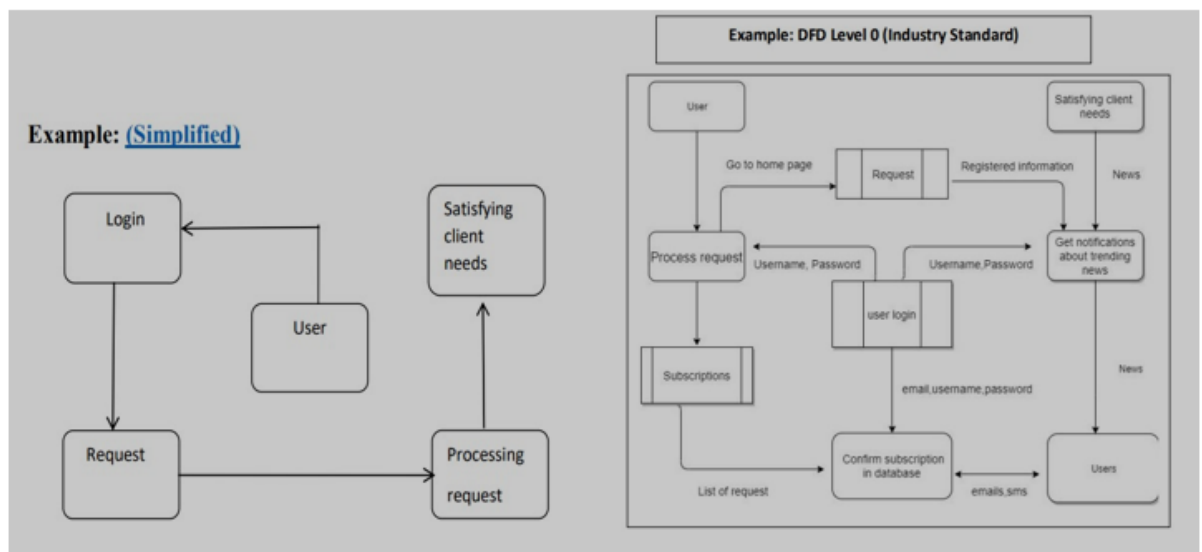
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The NEWS tracker app has simple interface so that all people can access it easily. User can access favourite news and magazine. User gets instant notification on interested topic.
NFR-2	Security	To ensure the security of the user related data and app data.
NFR-3	Reliability	How quick the user issues are resolved.
NFR-4	Performance	How quick the app responds for user inputs and how quickly the issues are rectified.

NFR-5	Availability	App provides instant news feeds and in how many languages news feeds are available.
NFR-6	Scalability	How the app is updated for the future needs without compromising the performance and improve security features in future.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

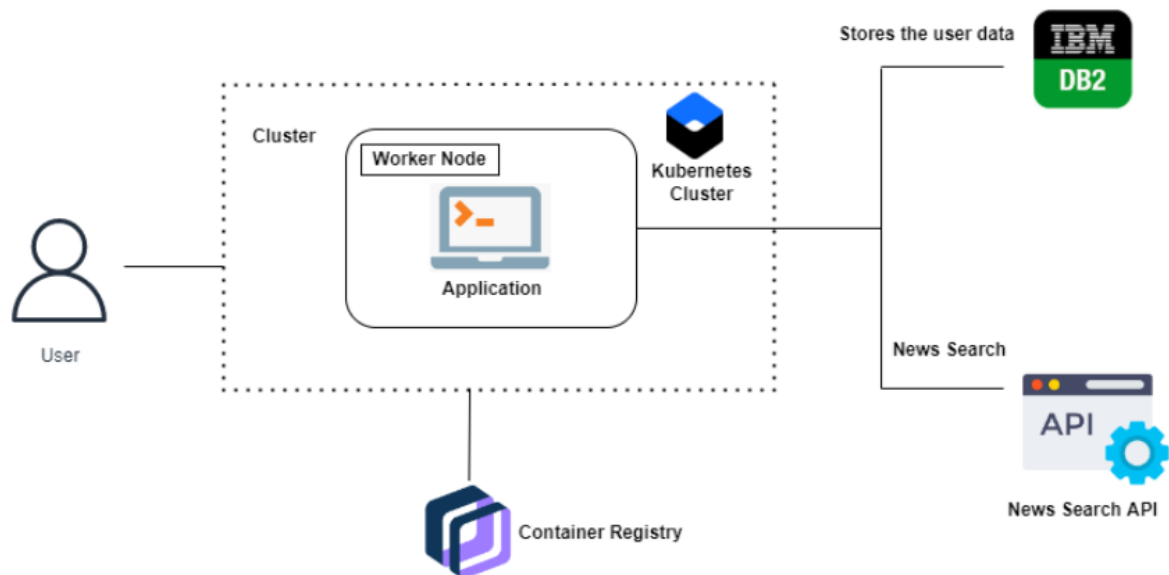
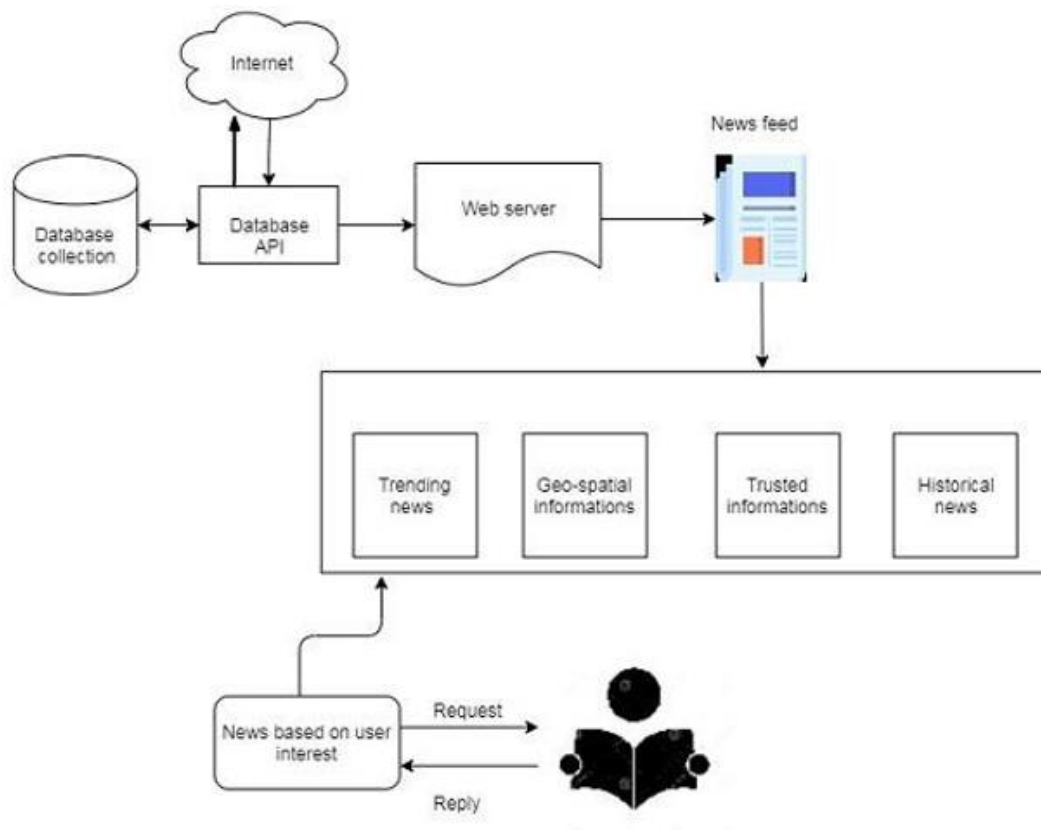
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

1. Find the best tech solution to solve existing business problems.
2. Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
3. Define features, development phases, and solution requirements.
4. Provide specifications according to which the solution is defined, managed, and delivered.



5.3 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Searching news)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm.	High	Sprint-1

		USN-3	As a user, I can register for the application through their given website.	I can register & access the dashboard with Gmail or in Browser Login.	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail.		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password.	I can view all types of informations through this application	High	Sprint-1
	Dashboard	USN-6	To see their histories about recently viewed, updates for search related news, current progress ,feedback.			

	Browser	USN-7	Have interactive medium between client and server.	I have a clarity to use this application and easily resolve my specific issues	High	Sprint-1
	Chat Bot	USN-8	Rectify the issues related subscription, account ,terms and conditions, privacy policy.			

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

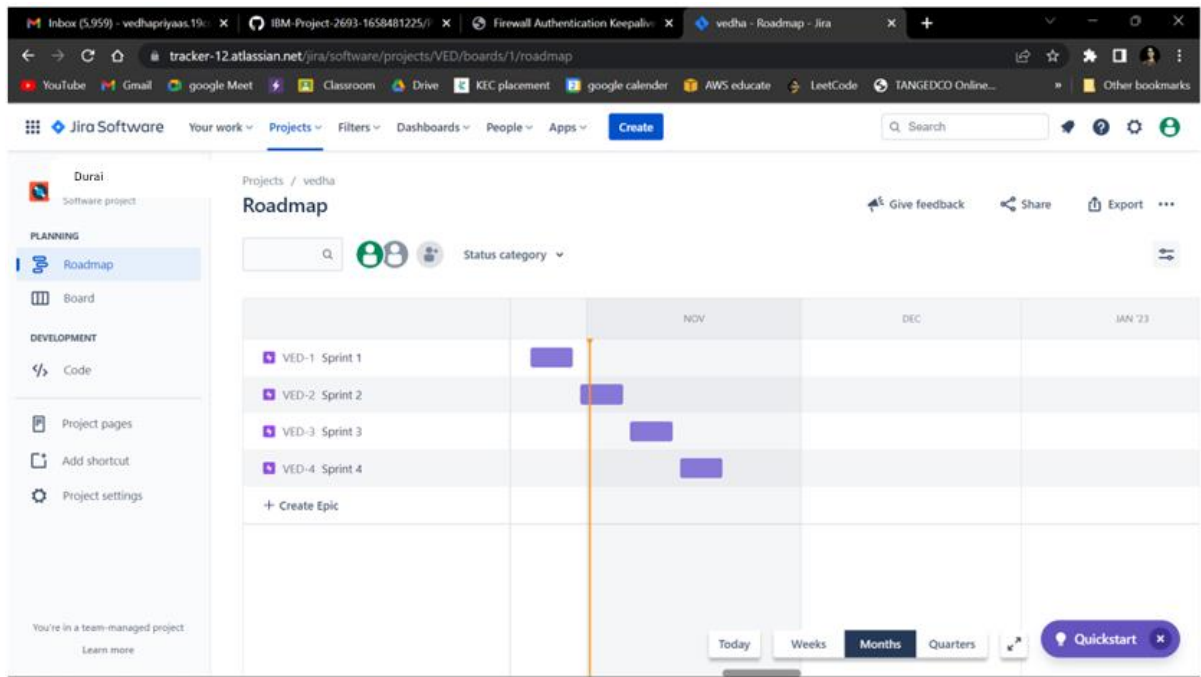
$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	S.Durai Raj, R.Sivanesan, R.Sathish Kumar, M.Shivaram.
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	10	High	R.Sivanesan, S.Durai Raj, R.Sathish Kumar, M.Shivaram.
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password.	15	High	R.Sathish Kuma S.Durai Raj, R.Sivanesan, M.Shivaram.

Sprint-2	Input Necessary Details	USN-4	As a user, I can search the news in the application	15	High	M.Shivaram S.Durai Raj, R.Sivanesan, R.Sathish Kumar.
Sprint-2	Data Pre-processing	USN-5	The application searches for news related to the entered details.	15	High	M.Shivaram S.Durai Raj, R.Sivanesan, R.Sathish Kumar.
Sprint-3	Searching of news	USN-6	As a user, I can search for the accurate news what I want	20	High	S.Durai Raj R.Sivanesan, R.Sathish Kumar, M.Shivaram.
Sprint-3		USN-7	As a user, I can get accurate news in the application	5	Medium	R.Sathish Kumar, S.Durai Raj, R.Sivanesan, M.Shivaram.
Sprint-4	Review	USN-8	As a user, I can give feedback of the application.	20	High	S.Durai Raj, R.Sivanesan, R.Sathish Kumar, M.Shivaram.

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature login & Signup:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    font-family: 'Jost', sans-serif;
    background: linear-gradient(to bottom, black, #302b63, #24243e);
  }
  .main{
    width: 350px;
    height: 500px;
    background: red;
    overflow: hidden;
```

```
background: url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dm
is5/fo0picsp1nhiucmc0l25s29respgr4j/1631524275000/035223609609222983
74/03522360960922298374/1Sx0jhdpEpnNIydS4rnN4kHSJtU1EyWka?e=view
&authuser=0&nonce=gcrocepgbb17m&user=03522360960922298374&hash=tf
hgbs86ka6divo3llbvp93mg4csvb38") no-repeat center/ cover;
border-radius: 10px;
box-shadow: 5px 20px 50px #000;
}
#chk{
display: none;
}
.signup{
position: relative;
width: 100%;
height: 100%;
}
label{
color: #fff;
font-size: 2.3em;
justify-content: center;
display: flex;
margin: 60px;
font-weight: bold;
cursor: pointer;
transition: .5s ease-in-out;
}
input{
width: 60%;
height: 20px;
background: #e0dede;
justify-content: center;
display: flex;
margin: 20px auto;
padding: 10px;
border: none;
outline: none;
border-radius: 5px;
}
button{
width: 60%;
height: 40px;
margin: 10px auto;
```

```
justify-content: center;
display: block;
color: #fff;
background: #573b8a;
font-size: 1em;
font-weight: bold;
margin-top: 20px;
outline: none;
border: none;
border-radius: 5px;
transition: .2s ease-in;
cursor: pointer;
}
button:hover{
  background: #6d44b8;
}
.login{
  height: 460px;
  background: #eee;
  border-radius: 60% / 10%;
  transform: translateY(-180px);
  transition: .8s ease-in-out;
}
.login label{
  color: #573b8a;
  transform: scale(.6);
}

#chk:checked ~ .login{
  transform: translateY(-500px);
}
#chk:checked ~ .login label{
  transform: scale(1);
}
#chk:checked ~ .signup label{
  transform: scale(.6);
}

</style>
<title>Login/Signup</title>
<link rel="stylesheet" type="text/css" href="slide navbar style.css">
```

```

<link
href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
</head>
<body>
    <div class="main">
        <input type="checkbox" id="chk" aria-hidden="true">

            <div class="signup">
                <form>
                    <label for="chk" aria-hidden="true">Sign
up</label>
                    <input type="text" name="txt"
placeholder="User name" required="">
                    <input type="email" name="email"
placeholder="Email" required="">
                    <input type="password" name="pswd"
placeholder="Password" required="">
                    <button><a href="/base">Sign
up</a></button>
                </form>
            </div>

            <div class="login">
                <form>
                    <label for="chk" aria-
hidden="true">Login</label>
                    <input type="email" name="email"
placeholder="Email" required="">
                    <input type="password" name="pswd"
placeholder="Password" required="">
                    <button><a href="/base">Login</a></button>
                </form>
            </div>
        </div>
    </body>
</html>

```

Base:

```

<!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/twitter-
bootstrap/4.6.0/css/bootstrap.min.css">
<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')
}}">
<title>Daily Bytes</title>
</head>
<body>

    {% block navbar %}

    <div class="container-fluid" style="background-color: #000;">
        <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top
shadow-sm">

            <!-- Show this only on mobile to medium screens -->
            <a class="navbar-brand d-lg-none" href="{{ url_for('home') }}"><b
class="lg"><span class="logo">Daily</span> Bytes</b></a>
            <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarToggle" aria-controls="navbarToggle" aria-
expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <!-- Use flexbox utility classes to change how the child elements are
justified -->
            <div class="collapse navbar-collapse justify-content-between"
id="navbarToggle">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="{{ url_for('headlines')
}}">Headlines</a>
                    </li>
                    <li class="nav-item">

```

```

        <a class="nav-link"href="{ { url_for('articles') } }">Articles</a>
    </li>
    <li class="nav-item">
        <a class="nav-link"href="{ { url_for('sources') } }">Sources</a>
    </li>
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdown" role="button" data-toggle="dropdown" aria-
expanded="false">
            Categories
        </a>
        <div class="dropdown-menu" aria-
labelledby="navbarDropdown">
            <a class="dropdown-item" href="{ { url_for('business')
} }">Business</a>
            <a class="dropdown-item" href="{ { url_for('tech')
} }">Technology</a>
            <a class="dropdown-item" href="{ { url_for('entertainment')
} }">Entertainment</a>
            <a class="dropdown-item" href="{ { url_for('science')
} }">Science</a>
            <a class="dropdown-item" href="{ { url_for('sports')
} }">Sport</a>
            <a class="dropdown-item" href="{ { url_for('health')
} }">Health</a>
        </div>
    </li>
</ul>
<!-- Show this only lg screens and up -->
    <a class="navbar-brand d-none d-lg-block" href="{ { url_for('home')
} }"><b class="lg"><span class="logo">Daily</span> Bytes</b></a>
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link" href="#">
                <svg class="svg" xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,
1);"><path d="M13 7h-2v4H7v2h4v4h2v-4h4v-2h-4z"></path><path d="M12
2C6.486 2 2 6.486 2 12s4.486 10 10 10-4.486 10-10S17.514 2 12 2zm0 18c-
4.411 0-8-3.589-8-8s3.589-8 8-8 3.589 8 8-3.589 8-8 8z"></path></svg>
            </a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">

```

```
<svg class="svg" xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,
1);"><path d="M16.5,8c0,1.5-0.5,3.5-2.9,4.3c0.7-1.7,0.8-3.4,0.3-5c-0.7-2.1-3-
3.7-4.6-4.6C8.9,2.4,8.2,2.8,8.3,3.4c0,1.1-0.3,2.7-
2,4.4 C4.1,10,3,12.3,3,14.5C3,17.4,5,21,9,21c-4-4-1-7.5-1-
7.5c0.8,5.9,5,7.5,7,7.5c1.7,0,5-1.2,5-6.4c0-3.1-1.3-5.5-2.4-
6.9 C17.3,7.2,16.6,7.5,16.5,8"></path></svg>
```

```
</a>
```

```
</li>
```

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

```
<a class="nav-link" href="#">
```

```
<svg class="svg" xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,
1);"><path d="M19 13.586V10c0-3.217-2.185-5.927-5.145-6.742C13.562 2.52
12.846 2 12 2s-1.562,5.2-1.855 1.258C7.185 4.074 5 6.783 5 10v3.586l-1.707
1.707A.996.996 0 0 0 3 16v2a1 1 0 0 0 1 1h16a1 1 0 0 0 1-1v-2a.996.996 0 0 0-
.293-.707L19 13.586zM19 17H5v-.586l1.707-1.707A.996.996 0 0 0 7 14v-4c0-
2.757 2.243-5 5-5s5 2.243 5 5v4c0 .266.105.52.293.707L19 16.414V17zm-7
5a2.98 2.98 0 0 0 2.818-2H9.182A2.98 2.98 0 0 0 12 22z"></path></svg>
```

```
</a>
```

```
</li>
```

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

```
<a class="nav-link" href="logout">
```

```
<svg class="svg" xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 24 24" width="24" height="24" style="fill: rgba(255, 255, 255,
1);"><path d="M12 2A10.13 10.13 0 0 0 2 12a10 10 0 0 0 4 7.92V20h.1a9.7
9.7 0 0 0 11.8 0h.1v-.08A10 10 0 0 0 22 12 10.13 10.13 0 0 0 12 2zM8.07
18.93A3 3 0 0 1 11 16.57h2a3 3 0 0 1 2.93 2.36 7.75 7.75 0 0 1-7.86 0zm9.54-
1.29A5 5 0 0 0 13 14.57h-2a5 5 0 0 0-4.61 3.07A8 8 0 0 1 4 12a8.1 8.1 0 0 1 8-
8 8.1 8.1 0 0 1 8 8 8 0 0 1-2.39 5.64z"></path><path d="M12 6a3.91 3.91 0 0
0-4 4 3.91 3.91 0 0 0 4 4 3.91 3.91 0 0 0 4-4 3.91 3.91 0 0 0-4-4zm0 6a1.91
1.91 0 0 1-2-2 1.91 1.91 0 0 1 2-2 1.91 1.91 0 0 1 2 2 1.91 1.91 0 0 1-2
2z"></path></svg> Logout
```

```
</a>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
</nav>
```

```
</div>
```

```
{% endblock % }
```

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



```

{% endblock % }

{% block footer % }

<div class="container-fluid footer">
  <p class="text-center footer-text">
    © <span class="logo">News</span>
  </p>
</div>

{% endblock % }

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

```

7.2 Feature 2

Home:

```

{% extends 'base.html' % }

{% block content % }

<div class="container-fluid landing">
  <div class="row">
    {% for source, title, desc, author, img, p_date, url in articles % }
    <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);"><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);"><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 6-6-6z"></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:

```

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
    <div class="row">
        {% for source, title, desc, author, img, p_date, url in headlines %}
        <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);"><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);"><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 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 %}

```

Articles:

```

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
    <div class="row">
        {% for source, title, desc, author, img, p_date, url in random %}
        <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);"><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);"><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 %}
```

Entertainment:

```
{% 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);"><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);"><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 %}

```

Business:

```

{% 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);"><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);"><path d="M12 5c-4.411 0-8 3.589-
8 8s3.589 8 8 8-3.589 8-8 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 %}

```

Sports:

```

{% 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>
                </div>
            </div>
        </div>
        </div>
    </div>

```

```

        <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);"><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);"><path d="M12 5c-4.411 0-8 3.589-
8 8s3.589 8 8 8-3.589 8-8 8-8-3.589-8-8zm0 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 %}

```

8.TESTING:

Test case	feature	component	Test scenario	Expected result	Actual result	status	comments	bug	Executed by
Sign in	Functional	Login page	Verify user can see the <u>sign in</u> option	can visible	Yes visible	pass	successful	-	Sathish kumar
Sign up	Functional	<u>Login page</u>	Verify user has the option to sign up	Can visible	Yes visible	pass	Successful	-	Sivanesan

News feeds	Functional	Home page	Verify user can get the news	News will be feed to the app	Feeds are appearing	yes	successful	-	Durai
Types of news available in the categories options	Functional	Dashboard	Types of news available	Weather, Sport, Economy.	Hover buttons will be shown.	Yes	successful	-	Shivaram

9.RESULT:

News tracker application using cloud is developed and executed at the level of completed progress .

10. ADVANTAGES & DISADVANTAGES

Advantages:-

- ❖ In this app news is already categorization.
- ❖ 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.
- ❖ Occurance of Advertisement disturb the user.
- ❖ Sometimes the news gives brief information.
- ❖ Prevalance of fake and uncertain news can confuse the user 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

main.py:

```
from app import app
if __name__ == "__main__":
    app.run()
```

init.py:

```
from flask import Flask
app = Flask(__name__)
from app import views
```

config.py:

```
class Config:

    NEWS_BASE_URL_SOURCES =
'https://newsapi.org/v2/everything?domains={ }&apiKey={ }'
    NEWS_BASE_EVERYTHING_URL =
'https://newsapi.org/v2/everything?domains={ }&apiKey={ }'
    NEWS_BASE_HEADLINES_URL = 'https://newsapi.org/v2/top-
headlines?country=us&apiKey={ }'
    NEWS_BASE_SOURCE = 'https://newsapi.org/v2/top-
headlines?sources={ }&apiKey={ }'
    API_KEY = "26b61e1de89f4d829e3b925f347b91a7"

class ProdConfig(Config):
    pass

class DevConfig(Config):
    DEBUG = True

config_options= {
    'development': DevConfig,
    'production': ProdConfig
}
```

models.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 .models import Articles
from .models import Sources
from newsapi import NewsApiClient
from .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

def publishedArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)

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

    all_articles = get_articles['articles']

    articles_results = []

    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)

    articles_results.append(article_object)

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

return contents

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']

    articles_results = []

    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_headlines)):
        headline = all_headlines[i]

```

```

source.append(headline['source'])
title.append(headline['title'])
desc.append(headline['description'])
author.append(headline['author'])
img.append(headline['urlToImage'])
p_date.append(headline['publishedAt'])
url.append(headline['url'])

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

articles_results.append(article_object)

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

return contents

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']

    articles_results = []

    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)

        articles_results.append(article_object)

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

    return contents

def businessArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)

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

    all_articles = business_articles['articles']

    business_articles_results = []

    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)

        business_articles_results.append(article_object)

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

    return contents

```

```

def techArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)

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

    all_articles = tech_articles['articles']

    tech_articles_results = []

    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)

        tech_articles_results.append(article_object)

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

    return contents

def entArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)

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

    all_articles = ent_articles['articles']

    ent_articles_results = []

```

```

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)

    ent_articles_results.append(article_object)

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

return contents

```

```

def scienceArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)

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

    all_articles = science_articles['articles']

    science_articles_results = []

    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)

        science_articles_results.append(article_object)

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

    return contents

def sportArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)

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

    all_articles = sport_articles['articles']

    sport_articles_results = []

    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)

        sport_articles_results.append(article_object)

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

    return contents

def healthArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)

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

    all_articles = health_articles['articles']

    health_articles_results = []

    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)

        health_articles_results.append(article_object)

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

    return contents

```

```

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 render_template
from .request import businessArticles, entArticles, get_news_source,
healthArticles, publishedArticles, randomArticles, scienceArticles, sportArticles,
techArticles, topHeadlines

```

```

@app.route('/base')

```

```
def base():
    articles = publishedArticles()
    return render_template('base.html', articles = articles)

@app.route('/home')
def home():
    articles = publishedArticles()
    return render_template('home.html', articles = articles)

@app.route('/')
def signin():
    articles = publishedArticles()
    return render_template('signin.html', articles = articles)

@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)

@app.route('/register')
def register():
    return render_template('register.html')
@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/logout')
def logout():
    return render_template('logout.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 articles % }
    <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);"><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);"><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>
    { % endfor % }
</div>

{ % endblock % }

```

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/twitter-bootstrap/4.6.0/css/bootstrap.min.css">
<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') }}">
<title>Daily Bytes</title>
</head>
<body>

    {% block navbar %}

        <div class="container-fluid" style="background-color: #000;">
            <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top shadow-sm">

                <!-- Show this only on mobile to medium screens -->
                <a class="navbar-brand d-lg-none" href="{{ url_for('home') }}"><b class="lg"><span class="logo">Daily</span> Bytes</b></a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarToggle" aria-controls="navbarToggle" aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <!-- Use flexbox utility classes to change how the child elements are justified -->
                <div class="collapse navbar-collapse justify-content-between" id="navbarToggle">
                    <ul class="navbar-nav">
                        <li class="nav-item">
                            <a class="nav-link" href="{{ url_for('headlines') }}">Headlines</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="{{ url_for('articles') }}">Articles</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="{{ url_for('sources') }}">Sources</a>

```

```

        </li>
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdown" role="button" data-toggle="dropdown" aria-
expanded="false">
                Categories
            </a>
            <div class="dropdown-menu" aria-
labelledby="navbarDropdown">
                <a class="dropdown-item" href="{ { url_for('business')
}}">Business</a>
                <a class="dropdown-item" href="{ { url_for('tech')
}}">Technology</a>
                <a class="dropdown-item" href="{ { url_for('entertainment')
}}">Entertainment</a>
                <a class="dropdown-item" href="{ { url_for('science')
}}">Science</a>
                <a class="dropdown-item" href="{ { url_for('sports')
}}">Sport</a>
                <a class="dropdown-item" href="{ { url_for('health')
}}">Health</a>
            </div>
        </li>
    </ul>
    <!-- Show this only lg screens and up -->
    <a class="navbar-brand d-none d-lg-block" href="{ { url_for('home')
}}"><b class="lg"><span class="logo">Daily</span> Bytes</b></a>
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link" href="#">
                <svg class="svg" xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,
1);"><path d="M13 7h-2v4H7v2h4v4h2v-4h4v-2h-4z"></path><path d="M12
2C6.486 2 2 6.486 2 12s4.486 10 10 10-4.486 10-10S17.514 2 12 2zm0 18c-
4.411 0-8-3.589-8-8s3.589-8 8-8 3.589 8 8-3.589 8-8 8z"></path></svg>
            </a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">
                <svg class="svg" xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255,
1);"><path d="M16.5,8c0,1.5-0.5,3.5-2.9,4.3c0.7-1.7,0.8-3.4,0.3-5c-0.7-2.1-3-
3.7-4.6-4.6C8.9,2.4,8.2,2.8,8.3,3.4c0,1.1-0.3,2.7-

```

2,4.4 C4.1,10,3,12.3,3,14.5C3,17.4,5,21,9,21c-4-4-1-7.5-1-7.5c0.8,5.9,5,7.5,7,7.5c1.7,0,5-1.2,5-6.4c0-3.1-1.3-5.5-2.4-6.9 C17.3,7.2,16.6,7.5,16.5,8"></path></svg>

<li class="nav-item">

<svg class="svg" xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" style="fill: rgba(255, 255, 255, 1);"><path d="M19 13.586V10c0-3.217-2.185-5.927-5.145-6.742C13.562 2.52 12.846 2 12 2s-1.562.52-1.855 1.258C7.185 4.074 5 6.783 5 10v3.586l-1.707 1.707A.996.996 0 0 0 3 16v2a1 1 0 0 0 1 1h16a1 1 0 0 0 1 1v-2a.996.996 0 0 0-.293-.707L19 13.586zM19 17H5v-.586l1.707-1.707A.996.996 0 0 0 7 14v-4c0-2.757 2.243-5 5-5s 2.243 5 5v4c0 .266.105.52.293.707L19 16.414V17zm-7 5a2.98 2.98 0 0 0 2.818-2H9.182A2.98 2.98 0 0 0 12 22z"></path></svg>

<li class="nav-item">

<svg class="svg" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="24" height="24" style="fill: rgba(255, 255, 255, 1);"><path d="M12 2A10.13 10.13 0 0 0 2 12a10 10 0 0 0 4 7.92V20h.1a9.7 9.7 0 0 0 11.8 0h.1v-.08A10 10 0 0 0 22 12 10.13 10.13 0 0 0 12 2zM8.07 18.93A3 3 0 0 1 11 16.57h2a3 3 0 0 1 2.93 2.36 7.75 7.75 0 0 1-7.86 0zm9.54-1.29A5 5 0 0 13 14.57h-2a5 5 0 0 0-4.61 3.07A8 8 0 0 1 4 12a8.1 8.1 0 0 1 8-8 8.1 8.1 0 0 1 8 8 8 8 0 0 1-2.39 5.64z"></path><path d="M12 6a3.91 3.91 0 0 0-4 4 3.91 3.91 0 0 0 4 4 3.91 3.91 0 0 0 4-4zm0 6a1.91 1.91 0 0 1-2-2 1.91 1.91 0 0 1 2-2 1.91 1.91 0 0 1 2 2 1.91 1.91 0 0 1-2 2z"></path></svg>

Logout

</div>

</nav>

</div>

{% endblock % }

{% block content % }

{% endblock % }

{% block footer % }


```

<div class="container-fluid footer">
  <p class="text-center footer-text">
    © <span class="logo">News</span>
  </p>
</div>

{% endblock %}

<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.slim.min.js"></s
cript>
<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>
Login.Html:

```

```

<!DOCTYPE html>
<html>
<head>
<style>
body{
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  font-family: 'Jost', sans-serif;
  background: linear-gradient(to bottom, black, #302b63, #24243e);
}
.main{
  width: 350px;
  height: 500px;
  background: red;
  overflow: hidden;
  background:
    url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dm
is5/fo0picsp1nhiucmc0l25s29respgpr4j/1631524275000/035223609609222983
74/03522360960922298374/1Sx0jhdPepnNIydS4rnN4kHSJtU1EyWka?e=view

```

&authuser=0&nonce=gcrocepgbb17m&user=03522360960922298374&hash=tfhgbs86ka6divo3llbvp93mg4csvb38") no-repeat center/ cover;

```
border-radius: 10px;
box-shadow: 5px 20px 50px #000;
}
```

```
#chk{
display: none;
}
```

```
.signup{
position: relative;
width: 100%;
height: 100%;
}
```

```
label{
color: #fff;
font-size: 2.3em;
justify-content: center;
display: flex;
margin: 60px;
font-weight: bold;
cursor: pointer;
transition: .5s ease-in-out;
}
```

```
input{
width: 60%;
height: 20px;
background: #e0dede;
justify-content: center;
display: flex;
margin: 20px auto;
padding: 10px;
border: none;
outline: none;
border-radius: 5px;
}
```

```
button{
width: 60%;
height: 40px;
margin: 10px auto;
justify-content: center;
display: block;
color: #fff;
background: #573b8a;
```

```

font-size: 1em;
font-weight: bold;
margin-top: 20px;
outline: none;
border: none;
border-radius: 5px;
transition: .2s ease-in;
cursor: pointer;
}
button:hover{
    background: #6d44b8;
}
.login{
    height: 460px;
    background: #eee;
    border-radius: 60% / 10%;
    transform: translateY(-180px);
    transition: .8s ease-in-out;
}
.login label{
    color: #573b8a;
    transform: scale(.6);
}

#chk:checked ~ .login{
    transform: translateY(-500px);
}
#chk:checked ~ .login label{
    transform: scale(1);
}
#chk:checked ~ .signup label{
    transform: scale(.6);
}

</style>
<title>Login/Signup</title>
<link rel="stylesheet" type="text/css" href="slide navbar style.css">
<link
href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
</head>
<body>
    <div class="main">

```

```

        <input type="checkbox" id="chk" aria-hidden="true">

        <div class="signup">
            <form>
                <label for="chk" aria-hidden="true">Sign
up</label>
                <input type="text" name="txt"
placeholder="User name" required="">
                <input type="email" name="email"
placeholder="Email" required="">
                <input type="password" name="pswd"
placeholder="Password" required="">
                <button><a href="/base">Sign up</a></button>
            </form>
        </div>

        <div class="login">
            <form>
                <label for="chk" aria-
hidden="true">Login</label>
                <input type="email" name="email"
placeholder="Email" required="">
                <input type="password" name="pswd"
placeholder="Password" required="">
                <button><a href="/base">Login</a></button>
            </form>
        </div>
    </div>
</body>
</html>

```

Signup.Html:

```

<!DOCTYPE html>
<html>
<head>
<style>
    body{
        margin: 0;
        padding: 0;
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
        font-family: 'Jost', sans-serif;
    }

```

```

    background: linear-gradient(to bottom, #0f0c29, #302b63, #24243e);
}
.main{
    width: 350px;
    height: 500px;
    background: red;
    overflow: hidden;
    background: url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dm
is5/fo0picsp1nhiucmc0l25s29respgpr4j/1631524275000/035223609609222983
74/03522360960922298374/1Sx0jhdpEpnNIydS4rnN4kHSJtU1EyWka?e=view
&authuser=0&nonce=gcrocepbb17m&user=03522360960922298374&hash=tf
hgbs86ka6divo3llbvp93mg4csvb38") no-repeat center/ cover;
    border-radius: 10px;
    box-shadow: 5px 20px 50px #000;
}
#chk{
    display: none;
}
.signup{
    position: relative;
    width: 100%;
    height: 100%;
}
label{
    color: #fff;
    font-size: 2.3em;
    justify-content: center;
    display: flex;
    margin: 60px;
    font-weight: bold;
    cursor: pointer;
    transition: .5s ease-in-out;
}
input{
    width: 60%;
    height: 20px;
    background: #e0dede;
    justify-content: center;
    display: flex;
    margin: 20px auto;
    padding: 10px;
    border: none;
}

```

```
outline: none;
border-radius: 5px;
}
button{
width: 60%;
height: 40px;
margin: 10px auto;
justify-content: center;
display: block;
color: #fff;
background: #573b8a;
font-size: 1em;
font-weight: bold;
margin-top: 20px;
outline: none;
border: none;
border-radius: 5px;
transition: .2s ease-in;
cursor: pointer;
}
button:hover{
background: #6d44b8;
}
.login{
height: 460px;
background: #eee;
border-radius: 60% / 10%;
transform: translateY(-180px);
transition: .8s ease-in-out;
}
.login label{
color: #573b8a;
transform: scale(.6);
}

#chk:checked ~ .login{
transform: translateY(-500px);
}
#chk:checked ~ .login label{
transform: scale(1);
}
#chk:checked ~ .signup label{
transform: scale(.6);
```

```

    }

</style>
    <title>Login/Signup</title>
    <link rel="stylesheet" type="text/css" href="slide navbar style.css">
<link
href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
</head>
<body>
    <div class="main">
        <input type="checkbox" id="chk" aria-hidden="true">

            <div class="signup">
                <form>
                    <label for="chk" aria-hidden="true">Sign
up</label>
                    <input type="text" name="txt"
placeholder="User name" required="">
                    <input type="email" name="email"
placeholder="Email" required="">
                    <input type="password" name="pswd"
placeholder="Password" required="">
                    <button ><a href="/base">Sign
up</a></button>
                </form>
            </div>

            <div class="login">
                <form>
                    <label for="chk" aria-
hidden="true">Login</label>
                    <input type="email" name="email"
placeholder="Email" required="">
                    <input type="password" name="pswd"
placeholder="Password" required="">
                    <button><a href="/base">Login</a></button>
                </form>
            </div>
        </div>
    </body>
</html>

```

Logout.Html:

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh;
  font-family: 'Jost', sans-serif;
  background: linear-gradient(to bottom, #0f0c29, #302b63, #24243e);
}
.main{
  width: 350px;
  height: 500px;
  background: red;
  overflow: hidden;
  background:
    url("https://doc-08-2c-
docs.googleusercontent.com/docs/securesc/68c90smiglihng9534mvqmq1946dm
is5/fo0picsp1nhiucmc0l25s29respgr4j/1631524275000/035223609609222983
74/03522360960922298374/1Sx0jhdpEpnNIydS4rnN4kHSJtU1EyWka?e=view
&authuser=0&nonce=gcrocepgbb17m&user=03522360960922298374&hash=tf
hgbs86ka6divo3llbvp93mg4csvb38") no-repeat center/ cover;
  border-radius: 10px;
  box-shadow: 5px 20px 50px #000;
}
#chk{
  display: none;
}
.signup{
  position: relative;
  width: 100%;
  height: 100%;
}
label{
  color: #fff;
  font-size: 2.3em;
  justify-content: center;
  display: flex;
  margin: 60px;
```



```
font-weight: bold;
cursor: pointer;
transition: .5s ease-in-out;
}
input{
width: 60%;
height: 20px;
background: #e0dede;
justify-content: center;
display: flex;
margin: 20px auto;
padding: 10px;
border: none;
outline: none;
border-radius: 5px;
}
button{
width: 60%;
height: 40px;
margin: 10px auto;
justify-content: center;
display: block;
color: #fff;
background: #573b8a;
font-size: 1em;
font-weight: bold;
margin-top: 20px;
outline: none;
border: none;
border-radius: 5px;
transition: .2s ease-in;
cursor: pointer;
}
button:hover{
background: #6d44b8;
}
.login{
height: 460px;
background: #eee;
border-radius: 60% / 10%;
transform: translateY(-180px);
transition: .8s ease-in-out;
}
```

```

.login label{
  color: #573b8a;
  transform: scale(.6);
}

#chk:checked ~ .login{
  transform: translateY(-500px);
}
#chk:checked ~ .login label{
  transform: scale(1);
}
#chk:checked ~ .signup label{
  transform: scale(.6);
}

</style>
<title>Login/Signup</title>
<link rel="stylesheet" type="text/css" href="slide navbar style.css">
<link
href="https://fonts.googleapis.com/css2?family=Jost:wght@500&display=swap"
rel="stylesheet">
</head>
<body>
  <div class="main">
    <input type="checkbox" id="chk" aria-hidden="true">

      <div class="signup">
        <form>
          <label for="chk" aria-hidden="true">Sign
up</label>
          <input type="text" name="txt"
placeholder="User name" required="">
          <input type="email" name="email"
placeholder="Email" required="">
          <input type="password" name="pswd"
placeholder="Password" required="">
          <button ><a href="/base">Sign
up</a></button>
        </form>
      </div>

      <div class="login">
        <form>

```

```

                                <label                for="chk"                aria-
hidden="true">Login</label>
                                <input                type="email"                name="email"
placeholder="Email" required="">
                                <input                type="password"                name="pswd"
placeholder="Password" required="">
                                <button><a href="/base">Login</a></button>
                                </form>
                            </div>
                    </div>
</body>
</html>

```

Articles.Html:

```

{% extends 'base.html' %}

{% block content %}

<div class="container-fluid landing">
    <div class="row">
        {% for source, title, desc, author, img, p_date, url in random %}
        <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);"><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);"><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 %}
```

Headline.Html:

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

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

```

<div class="container-fluid landing">
  <div class="row">
    {% for source, title, desc, author, img, p_date, url in headlines %}
    <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);"><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 3-1.726 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);"><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
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>
  </div>

```

```

        </div>
    </div>
</div>
{% endfor %}
</div>
</div>

{% endblock %}
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 articles %}
            <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);"><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);"><path d="M12 5c-4.411 0-8 3.589-8
8s3.589 8 8 3.589 8-3.589 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>
        {% endfor %}
    </div>
</div>

```

```
        </div>
    {% endfor %}
</div>
</div>
```

```
{% endblock %}
```

Business.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);"><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);"><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 % }
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);"><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);"><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 % }
```

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);"><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);"><path d="M12 5c-4.411 0-8 3.589-8
8s3.589 8 8 8-3.589 8-8 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 % }
```


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);"><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);"><path d="M12 5c-4.411 0-8 3.589-8
8s3.589 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 %}
```

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);"><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>
    </div>
  </div>
  {% endfor %}
</div>
```

```
{% endblock %}
```

Sports.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>
          </div>
        </div>
      </div>
    </div>
  </div>
```

```

        <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);"><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);"><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 %}
```

```
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>
                </div>
            </div>
        </div>
        </div>
    </div>

```

```

        <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);"><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);"><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>

{% endblock %}

```

Style.Css:

```

@import
url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display
=swap');

* {
    font-family:'Overpass', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #1e1e21;
    color: #fff;
}

p{
    font-size: 14px;
}

a {
    text-decoration: none;
}

```

```
    color: #fff;
}

a:hover {
    text-decoration: none;
    color: #fff;
}

.landing {
    margin-top: 100px;
}

#navbarToggle {
    background-color: #000;
}

.navbar.navbar-dark .nav-item .nav-link {
    color: #fff!important;
    background-color: #000;
}

.navbar.navbar-dark .nav-item .nav-link:hover {
    color: #fff!important;
    background-color: #000;
}

.navbar {
    background-color: #000!important;
}

.navbar-nav {
    background-color: #000;
}

.navbar-brand {
    background-color: #000;
}

.lg {
    background-color: #000;
    font-size: xx-large;
}

h1 {
```

```
    font-size: 33px;
    color: #dfdcd9;
}

h2 {
    font-size: 1.5em;
    color: #dfdcd9;
}

.author {
    font-size: 12px;
}

.date {
    font-size: 12px;
}

.article-card {
    position: relative;
    display: -ms-flexbox;
    display: flex;
    -ms-flex-direction: column;
    flex-direction: column;
    min-width: 0;
    color: #000;
    background-color: #fff;
    word-wrap: break-word;
    background-clip: border-box;
    border-radius: 10px;
    margin-top: 15px;
}

.img {
    width: 100%;
}

.image-fluid {
    height: 18em;
}

.cardfooter {
    padding: 0.75rem 1.25rem;
    padding-bottom: 0px;
    border-radius: 0px 0px 10px 10px;
```

```
}
```

```
.article_name {  
  color: #dfdcd9;  
}
```

```
.source-body {  
  margin-bottom: 30px;  
}
```

```
.source {  
  margin-top: 5px;  
  padding-top: 5px;  
  padding-bottom: 0px;  
  margin-bottom: 0px;  
  color: #dd0000;  
}
```

```
.logo {  
  color: #dd0000;  
  background-color: #000;  
}
```

```
.svg {  
  background-color: #000;  
}
```

```
.p-svg {  
  background-color: #26262a;  
}
```

```
.footer {  
  background-color: #000;  
  padding-top: 20px;  
  padding-bottom: 15px;  
}
```

```
.footer-text {  
  background-color: #000;  
  margin-bottom: 0px;  
}
```

```
.dropdown-menu {  
  background-color: #1e1e21;
```

```
}  
  
.dropdown-item {  
  color: #fff;  
}  
  
.dropdown-item:hover {  
  color: #fff;  
  background-color: #26262a;  
}  
  
.published {  
  background-color: #26262a;  
}  
  
.card-footer {  
  background-color: #26262a;  
}
```

PROJECT DEMO LINK:

<https://youtu.be/JqfDvjsy8>