

# REPORT

TEAM ID : PNT2022TMID01417

TEAM LEAD:M.VISHAL(211419205182)

TEAM MEMBER 1:Z.SYED IBRAHIM(211419205170)

TEAM MEMBER 2:S.NITHISH KUMAR(211419205119)

TEAM MEMBER 3:A.S.PRABU(211419205129)

# **ABSTRACT**

This Work is attempt to create and share the information to the people those who are in the busy world. The app should include all of the trustworthy sources from across the world and then specify in each article which source has validated this news because I'm assuming that customers can't discern the difference between real and fake news. This application is built with high data storage capacity,processing power and networking can all be scaled using cloud computing infrastructure.scaling can be done quickly and easily.

# **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**

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**

# **News Tracker Application**

## **1.INTRODUCTION**

Information regarding recent events is called news. Word-of-mouth, printing, postal services, radio, electronic communication, or the testimonies of observers and witnesses to the events may all be used to convey this. In order to set news apart from soft media, it is frequently referred to as "hard news." News reports frequently cover subjects like conflict, politics, education, health, the environment, business, fashion, entertainment, and sports, as well as odd or quirky occurrences. Since ancient times, government announcements on kingly rites, laws, taxes, public health, and criminals have been referred to as news. Government communication and espionage networks frequently promote technological and social advancements, which speed up the dissemination of news and affect its substance.

### **1.1 PROJECT OVERVIEW**

The necessity for customising news app interactions is clear as more people consume news on smartphones and tablets. We provide findings from three experiments that focus on crucial challenges in the design of adaptive news app interfaces. We started by asking people about their reading habits and interests. Our study identified three main categories of readers. After that, we developed and released an Android news app that keeps track of user interactions. We demonstrated that a classifier we trained using the logs can accurately identify a user based on the type of reader they are. Finally, we assessed various adaptive user interfaces for every type of reader. The review shows how different users of the news app will benefit from the modification differently, and it also shows that adaptive interfaces are feasible.

### **1.2 PURPOSE**

We frequently feel that we need more than 24 hours a day to do everything on our calendar because our lives are so hectic these days. That's not realistic, but you can cut down on the time by reading the news differently than you usually do. Simply let us know what market news you're interested in to receive a daily sneak glance. Save time by reading only the content you choose to be pertinent. With the aid of this app, you can search for all available data on indices, commodities, currencies, future rates, bonds, etc., just like on reputable websites.

## **2. LITERATURE SURVEY**

### **2.1 Existing problem**

We frequently believe that there aren't enough hours in the day to get everything done. That's not realistic, but you can cut down on the time by reading the news differently than you usually do.

### **2.2 References**

1.NEWS HUNTER: BUILDING AND MINING KNOWLEDGE GRAPHS FOR NEWSROOM SYSTEMS

Author: Arne Berven , Ole A. Christensen , Sindre Moldeklev , Andreas L. Opdahl, Kjetil J. Villanger

Year:2018

2.AI Algorithm System for Prediction of Diabetes Using Progressive Web Appand IBM Cloud.

Authors: Dr. Mohammed Abdul Raheem, Shaik Ehetesham, Mohammad Faiz Ahmed Subhani, Sayed Abdul Zakir

Year:2021

3.Online Privacy-Safe Engagement Tracking System

Author: Cheng Zhang,Cheng Chang,Lei Chen,Yang Liu

Year :2018

4.Using IBM Cloud services to support qualitative research

Author: EGZON KORENICA

Year:2021

5.A Cloud -based Architecture for Condition Monitoring based on Machine learning.

Year:2018

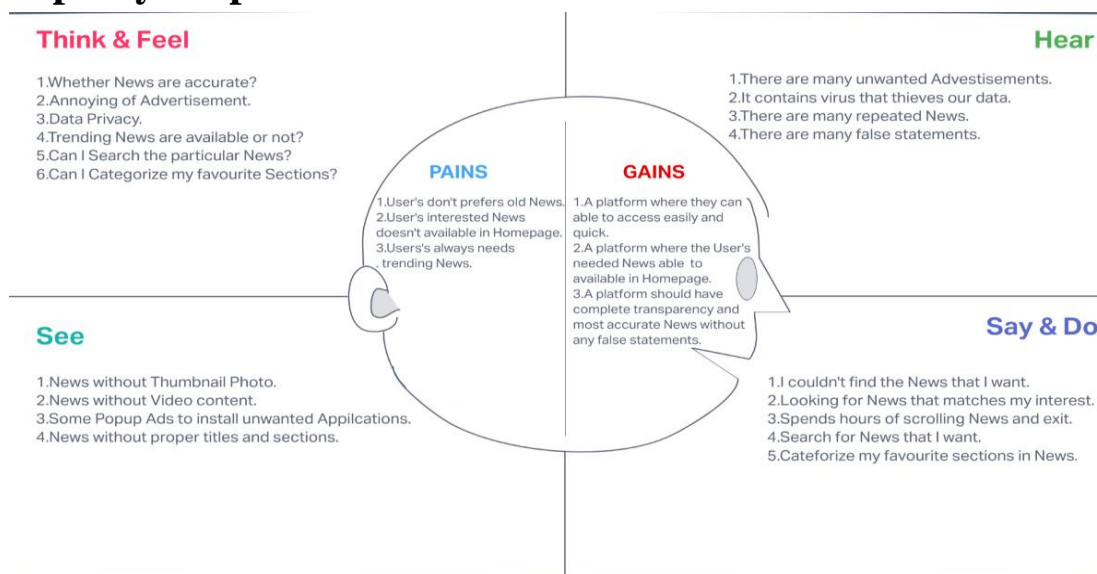
author: Fernando Arévalo; Mochammad Rizky Diprasetya; Andreas Schwung

## 2.3 Problem Statement Definition

Multiple news-sharing programmes might be used by one user, and these applications are regularly flooded with alerts. Furthermore, a lot of fake news is broadcast. A news-sharing app seeks to make it easier for users to acquire daily important and relevant news and to clearly indicate that the information is real and has been gathered from reputable sources.

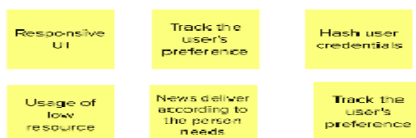
## 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas



### 3.2 Ideation & Brainstorming

#### Vishal



#### Syed



#### Prabu



#### Nithish



### 3.3 Proposed Solution

S.NO		PARAMETERS	DESCRIPTION
1		Problem Statement (Problem to be solved)	Multiple news-sharing programmes might be used by one user, and these applications are regularly flooded with alerts. Furthermore, a lot of fake news is broadcast. A news-sharing app seeks to make it easier for users to acquire daily important and relevant news and to clearly indicate that the information is real and has been gathered from reputable sources.
2		Idea / Solution description	The app should include all of the trustworthy sources from across the world and then specify in each article which source has validated this news because I'm assuming that customers can't discern the difference between real and fake news.
3		Novelty / Uniqueness	Creating attractive user interface for the customers and also deliver a important News Headliness with more details.
4		Social Impact / Customer Satisfaction	Creating Awareness about How to overcome negative thoughts and also spreading socialily responsible news.For customer Satisfication ,we Track all kind of news .it will be more Usefull to get knowledge in Today's World.
5		Business Model (Revenue Model)	Our First and most important Business is to deliver a good content or good news to the people apart from that We are going to promote advertisement about good products

6		Scalability of the Solution	This application is built with high data storage capacity ,processing power and networking can all be scaled using cloud computing infrastructure.scaling can be done quickly and easily.
---	--	-----------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 3.4 Problem Solution fit

Project Title: News Tracker Application

Project Design Phase-I - Solution Fit

Team ID: PNT2022TMD01417

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Working Professionals who doesn't have enough time to read the Newspaper.	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> Time and Budget.	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Newspaper is an alternative to Newstracker Application.  Pros: Users no longer need to read the news which they are not interested  Cons: Screen time may get increased.	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> 1. Reading Newspaper is time consuming task. 2. User may read uninterested news. 3. Regular buying of newspaper leads to exorbitant newspaper bills and may be result in unwanted scrap.	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> In olden days, User did not have enough internet facilities. So there is no other way than reading a newspaper, To know what's happening around!	<b>7. BEHAVIOUR</b> <span>BE</span> User need to follow the below steps: 1. Create an account in our webapp. 2. Login our webapp. 3. Categorize the news according to their interest.	
Identify strong TR & EM	<b>3. TRIGGERS</b> <span>TR</span> User when they see the neighbours stop buying Newspaper and subscribed to News Tracking Application.	<b>10. YOUR SOLUTION</b> <span>SL</span> We made this application is such a way that showing fake news in our application is impossible and we categorize the news according to the user interest which saves time for our busy users	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> 8.1 Online: User can categorize the news according to their interest and get notification  8.2 Offline: User can download the detailed news of the headlines and can read it offline	Identify strong TR & EM
	<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> Can see news only in television or newspaper > can see news anytime and anywhere just need your mobile phone			



## 4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login Via Email ID & Password
FR-4	Home Page	Shows the NEWS Headlines.
FR-5	Tracker Page	Shows the NEWS elaboratively when the NEWS Headline of the Home page is clicked.

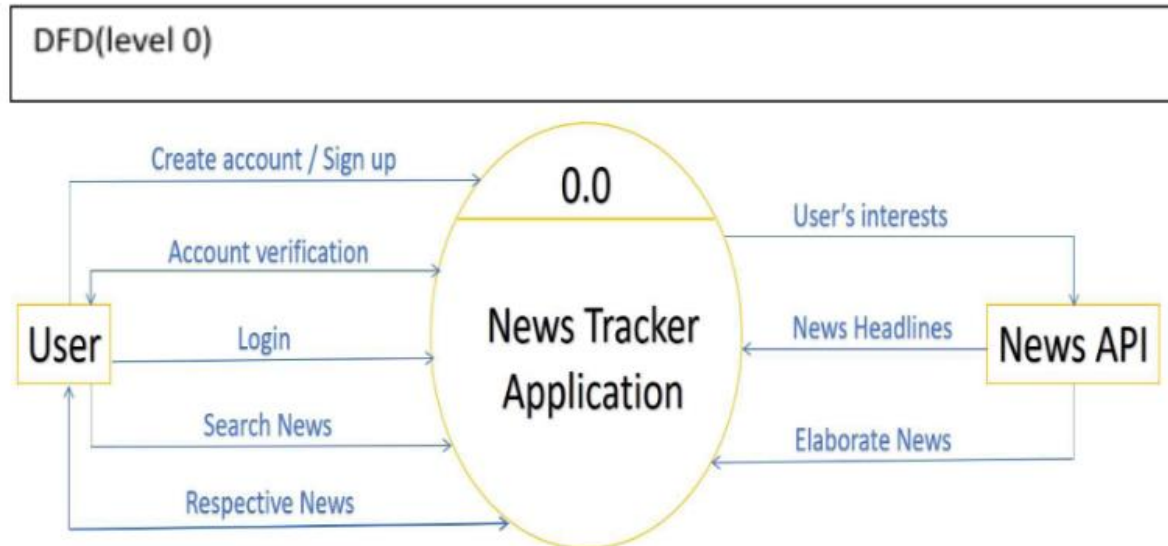
## 4.2 Non-functional Requirements:

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

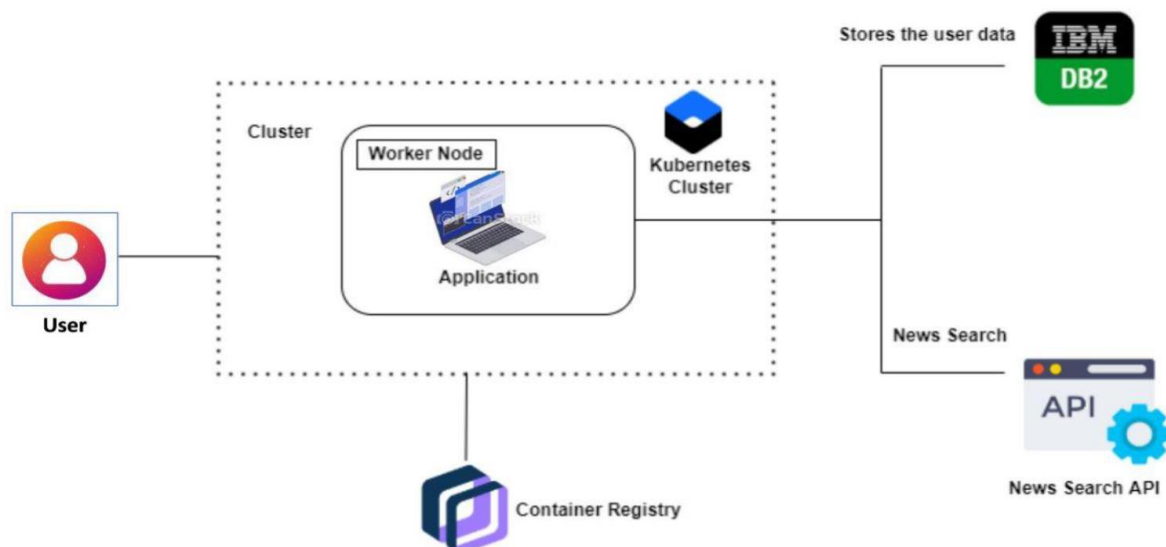
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Our Website is used to Read News and gain Knowledge according to user's interest.
NFR-2	Security	User can only login with their own Email ID and password without their Email and password no one can use the App.
NFR-3	Reliability	Our Website delivers the reliable content only from NEWS API.
NFR-4	Performance	It is a high performable App with friendly User Interface and shares lot of information without any Issues.
NFR-5	Availability	Our Website is supported with all search Engine so it will Avail for User at any time.
NFR-6	Scalability	This Website is built with high data storage capacity,processing power and networking can all be scaled using cloud computing. Infrastructure.scaling can be done quickly and easily.

## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams



### 5.2 Solution & Technical Architecture



## 5.3 User Stories

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	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
	Login	USN-3	As a user, I can log into the application by entering email & password	I can access my dashboard	High	Sprint-2
	Home Page	USN-4	As a user I can view the headlines of the news that interest me	I can read the news elaborately by clicking on the headlines	High	Sprint-2
		USN-5	As a user I can view the elaborate content of the headlines	I can download that to read it when iam free even in offline mode	Medium	Sprint-3
		USN-6	As a user I can search a news I want	I can read the news elaborately	High	Sprint-4
Customer Care Executive	Chatbot	USN-6	As a user I can solve my doubts about the application with the help of the chatbot.If the doubt is not resolved the chatbot guides us on how to contact the customer care executive	I can contact the customer care executive if needed	Medium	Sprint-3
Administrator	About us	USN-7	As a user I can view about the application and the developers of the application	I can be able to see their other applications	low	Sprint-4

## 6. PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

Title	Description	Date
Literature Survey	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	2nd September 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	7th September 2022

Ideation Brain Storming	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	15th September 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23rd September 2022
Problem Solution Fit	Prepare solution architecture document.	27th September 2022
Solution Architecture	Prepare solution architecture document.	30th September 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	5th October 2022
Solution Requirement	Prepare the functional requirement document	12th October 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	15th October 2022
Technology Architecture	Prepare the technology architecture diagram.	16th October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	21st October 2022

Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	
------------------------------------------------------	----------------------------------------------------	--

## 6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a user, I can log into the application by entering email & password	2	High	Vishal, Syed Ibrahim
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Nithish Kumar,prabu
Sprint-2		USN-3	As a user, I will receive confirmation email once I have registered for the application	1	High	Vishal,prabu
Sprint-2	Home Page	USN-4	As a user, I can view the headlines of the news that interest me	2	High	Nithish Kumar, Syed Ibrahim
Sprint-3		USN-5	As a user I can view the elaborate content of the headlines	2	Medium	Vishal,prabu
Sprint-4		USN-6	As a user I can search a news I want	2	High	Syed Ibrahim,prabu
Sprint-3	Chatbot	USN-7	As a user I can solve my doubts about the application with the help of the chatbot.if the doubt is not resolved the chatbot guides us on how to contact the customer care executive	2	Medium	Nithish Kumar,vishal
Sprint-4	About us	USN-8	As a user I can view about the application	1	Low	Vishal

## 7. CODING & SOLUTIONING

### SIGN UP.HTML

```
<!DOCTYPE html>
```

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

```
<head>
```

```
    <title>Sign up</title>
```

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

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

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

```
<link href="https://fonts.googleapis.com/css?family=Roboto:400,700"
rel="stylesheet">
```

```
<title>Bootstrap Simple Registration Form</title>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css">

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script
>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></sc
ript>

<script src="static/sign_up.js"></script>

<style>
    body {
        color: #fff;
        background: #63738a;
        font-family: 'Roboto', sans-serif;
    }
    .form-control{
        height: 40px;
        box-shadow: none;
        color: #969fa4;
    }
    .form-control:focus{
        border-color: #5cb85c;
    }
    .form-control, .btn{
        border-radius: 3px;
    }
}
```

```
.signup-form{
    width: 400px;
    margin: 0 auto;
    padding: 30px 0;
}

.signup-form h2{
    color: #636363;
margin: 0 0 15px;
    position: relative;
    text-align: center;
}

.signup-form h2:before, .signup-form h2:after{
    content: "";
    height: 2px;
    width: 30%;
    background: #d4d4d4;
    position: absolute;
    top: 50%;
    z-index: 2;
}

.signup-form h2:before{
    left: 0;
}

.signup-form h2:after{
    right: 0;
}

.signup-form .hint-text{
```



```
        color: #999;
        margin-bottom: 30px;
        text-align: center;
    }
.signup-form form{
    color: #999;
    border-radius: 3px;
    margin-bottom: 15px;
    background: #f2f3f7;
    box-shadow: 0px 2px 2px rgba(0, 0, 0, 0.3);
    padding: 30px;
}

.signup-form .form-group{
    margin-bottom: 20px;
}

.signup-form input[type="checkbox"]{
    margin-top: 3px;
}

.signup-form .btn{
font-size: 16px;
font-weight: bold;
    min-width: 140px;
    outline: none !important;
}

.signup-form .row div:first-child{
    padding-right: 10px;
}
```

```

        .signup-form .row div:last-child{
            padding-left: 10px;
        }
        .signup-form a{
            color: #fff;
            text-decoration: underline;
        }
        .signup-form a:hover{
            text-decoration: none;
        }
        .signup-form form a{
            color: #5cb85c;
            text-decoration: none;
        }
        .signup-form form a:hover{
            text-decoration: underline;
        }
</style>
</head>
<body>
<div class="signup-form">
    <form action="/sign_up" method="POST" name="signup" onsubmit="return
validateform()">
        <h2>Sign up</h2>
        <p class="hint-text">Create your account.</p>
        <div class="form-group">
            <input type="text" class="form-control" name="name"
placeholder="Name" id="username" required="required">

```

</div>

<div class="form-group">

<input type="email" class="form-control" name="email"  
placeholder="Email" id="email" required="required">

</div>

<div class="form-group">

<input type="phone" class="form-control" name="phone"  
placeholder="Phone No" id="phone" required="required">

</div>

<div class="form-group">

<select id="country" class="form-control" name="country"  
id="country">

<option value="select a country">---Select a country---</option>

<option value="Afghanistan">Afghanistan</option>

<option value="Åland Islands">Åland Islands</option>

<option value="Albania">Albania</option>

<option value="Algeria">Algeria</option>

<option value="American Samoa">American Samoa</option>

<option value="Andorra">Andorra</option>

<option value="Angola">Angola</option>

<option value="Anguilla">Anguilla</option>

<option value="Antarctica">Antarctica</option>

<option value="Antigua and Barbuda">Antigua and  
Barbuda</option>

<option value="Argentina">Argentina</option>

<option value="Armenia">Armenia</option>

<option value="Aruba">Aruba</option>

<option value="Australia">Australia</option>  
<option value="Austria">Austria</option>  
<option value="Azerbaijan">Azerbaijan</option>  
<option value="Bahamas">Bahamas</option>  
<option value="Bahrain">Bahrain</option>  
<option value="Bangladesh">Bangladesh</option>  
<option value="Barbados">Barbados</option>  
<option value="Belarus">Belarus</option>  
<option value="Belgium">Belgium</option>  
<option value="Belize">Belize</option>  
<option value="Benin">Benin</option>  
<option value="Bermuda">Bermuda</option>  
<option value="Bhutan">Bhutan</option>  
<option value="Bolivia">Bolivia</option>  
<option value="Bosnia and Herzegovina">Bosnia and  
Herzegovina</option>  
<option value="Botswana">Botswana</option>  
<option value="Bouvet Island">Bouvet Island</option>  
<option value="Brazil">Brazil</option>  
<option value="British Indian Ocean Territory">British Indian Ocean  
Territory</option>  
<option value="Brunei Darussalam">Brunei Darussalam</option>  
<option value="Bulgaria">Bulgaria</option>  
<option value="Burkina Faso">Burkina Faso</option>  
<option value="Burundi">Burundi</option>  
<option value="Cambodia">Cambodia</option>  
<option value="Cameroon">Cameroon</option>  
<option value="Canada">Canada</option>

<option value="Cape Verde">Cape Verde</option>  
<option value="Cayman Islands">Cayman Islands</option>  
<option value="Central African Republic">Central African  
Republic</option>  
<option value="Chad">Chad</option>  
<option value="Chile">Chile</option>  
<option value="China">China</option>  
<option value="Christmas Island">Christmas Island</option>  
<option value="Cocos (Keeling) Islands">Cocos (Keeling)  
Islands</option>  
<option value="Colombia">Colombia</option>  
<option value="Comoros">Comoros</option>  
<option value="Congo">Congo</option>  
<option value="Congo, The Democratic Republic of The">Congo,  
The Democratic Republic of The</option>  
<option value="Cook Islands">Cook Islands</option>  
<option value="Costa Rica">Costa Rica</option>  
<option value="Cote D'ivoire">Cote D'ivoire</option>  
<option value="Croatia">Croatia</option>  
<option value="Cuba">Cuba</option>  
<option value="Cyprus">Cyprus</option>  
<option value="Czech Republic">Czech Republic</option>  
<option value="Denmark">Denmark</option>  
<option value="Djibouti">Djibouti</option>  
<option value="Dominica">Dominica</option>  
<option value="Dominican Republic">Dominican Republic</option>  
<option value="Ecuador">Ecuador</option>  
<option value="Egypt">Egypt</option>

<option value="El Salvador">El Salvador</option>  
<option value="Equatorial Guinea">Equatorial Guinea</option>  
<option value="Eritrea">Eritrea</option>  
<option value="Estonia">Estonia</option>  
<option value="Ethiopia">Ethiopia</option>  
<option value="Falkland Islands (Malvinas)">Falkland Islands  
(Malvinas)</option>  
<option value="Faroe Islands">Faroe Islands</option>  
<option value="Fiji">Fiji</option>  
<option value="Finland">Finland</option>  
<option value="France">France</option>  
<option value="French Guiana">French Guiana</option>  
<option value="French Polynesia">French Polynesia</option>  
<option value="French Southern Territories">French Southern  
Territories</option>  
<option value="Gabon">Gabon</option>  
<option value="Gambia">Gambia</option>  
<option value="Georgia">Georgia</option>  
<option value="Germany">Germany</option>  
<option value="Ghana">Ghana</option>  
<option value="Gibraltar">Gibraltar</option>  
<option value="Greece">Greece</option>  
<option value="Greenland">Greenland</option>  
<option value="Grenada">Grenada</option>  
<option value="Guadeloupe">Guadeloupe</option>  
<option value="Guam">Guam</option>  
<option value="Guatemala">Guatemala</option>  
<option value="Guernsey">Guernsey</option>

<option value="Guinea">Guinea</option>

<option value="Guinea-bissau">Guinea-bissau</option>

<option value="Guyana">Guyana</option>

<option value="Haiti">Haiti</option>

<option value="Heard Island and Mcdonald Islands">Heard Island  
and Mcdonald Islands</option>

<option value="Holy See (Vatican City State)">Holy See (Vatican  
City State)</option>

<option value="Honduras">Honduras</option>

<option value="Hong Kong">Hong Kong</option>

<option value="Hungary">Hungary</option>

<option value="Iceland">Iceland</option>

<option value="India">India</option>

<option value="Indonesia">Indonesia</option>

<option value="Iran, Islamic Republic of">Iran, Islamic Republic  
of</option>

<option value="Iraq">Iraq</option>

<option value="Ireland">Ireland</option>

<option value="Isle of Man">Isle of Man</option>

<option value="Israel">Israel</option>

<option value="Italy">Italy</option>

<option value="Jamaica">Jamaica</option>

<option value="Japan">Japan</option>

<option value="Jersey">Jersey</option>

<option value="Jordan">Jordan</option>

<option value="Kazakhstan">Kazakhstan</option>

<option value="Kenya">Kenya</option>

<option value="Kiribati">Kiribati</option>

<option value="Korea, Democratic People's Republic of">Korea,  
Democratic People's Republic of</option>

<option value="Korea, Republic of">Korea, Republic of</option>

<option value="Kuwait">Kuwait</option>

<option value="Kyrgyzstan">Kyrgyzstan</option>

<option value="Lao People's Democratic Republic">Lao People's  
Democratic Republic</option>

<option value="Latvia">Latvia</option>

<option value="Lebanon">Lebanon</option>

<option value="Lesotho">Lesotho</option>

<option value="Liberia">Liberia</option>

<option value="Libyan Arab Jamahiriya">Libyan Arab  
Jamahiriya</option>

<option value="Liechtenstein">Liechtenstein</option>

<option value="Lithuania">Lithuania</option>

<option value="Luxembourg">Luxembourg</option>

<option value="Macao">Macao</option>

<option value="Macedonia, The Former Yugoslav Republic  
of">Macedonia, The Former Yugoslav Republic of</option>

<option value="Madagascar">Madagascar</option>

<option value="Malawi">Malawi</option>

<option value="Malaysia">Malaysia</option>

<option value="Maldives">Maldives</option>

<option value="Mali">Mali</option>

<option value="Malta">Malta</option>

<option value="Marshall Islands">Marshall Islands</option>

<option value="Martinique">Martinique</option>

<option value="Mauritania">Mauritania</option>

<option value="Mauritius">Mauritius</option>



<option value="Mayotte">Mayotte</option>

<option value="Mexico">Mexico</option>

<option value="Micronesia, Federated States of">Micronesia,  
Federated States of</option>

<option value="Moldova, Republic of">Moldova, Republic  
of</option>

<option value="Monaco">Monaco</option>

<option value="Mongolia">Mongolia</option>

<option value="Montenegro">Montenegro</option>

<option value="Montserrat">Montserrat</option>

<option value="Morocco">Morocco</option>

<option value="Mozambique">Mozambique</option>

<option value="Myanmar">Myanmar</option>

<option value="Namibia">Namibia</option>

<option value="Nauru">Nauru</option>

<option value="Nepal">Nepal</option>

<option value="Netherlands">Netherlands</option>

<option value="Netherlands Antilles">Netherlands Antilles</option>

<option value="New Caledonia">New Caledonia</option>

<option value="New Zealand">New Zealand</option>

<option value="Nicaragua">Nicaragua</option>

<option value="Niger">Niger</option>

<option value="Nigeria">Nigeria</option>

<option value="Niue">Niue</option>

<option value="Norfolk Island">Norfolk Island</option>

<option value="Northern Mariana Islands">Northern Mariana  
Islands</option>

<option value="Norway">Norway</option>

<option value="Oman">Oman</option>  
<option value="Pakistan">Pakistan</option>  
<option value="Palau">Palau</option>  
<option value="Palestinian Territory, Occupied">Palestinian  
Territory, Occupied</option>  
<option value="Panama">Panama</option>  
<option value="Papua New Guinea">Papua New Guinea</option>  
<option value="Paraguay">Paraguay</option>  
<option value="Peru">Peru</option>  
<option value="Philippines">Philippines</option>  
<option value="Pitcairn">Pitcairn</option>  
<option value="Poland">Poland</option>  
<option value="Portugal">Portugal</option>  
<option value="Puerto Rico">Puerto Rico</option>  
<option value="Qatar">Qatar</option>  
<option value="Reunion">Reunion</option>  
<option value="Romania">Romania</option>  
<option value="Russian Federation">Russian Federation</option>  
<option value="Rwanda">Rwanda</option>  
<option value="Saint Helena">Saint Helena</option>  
<option value="Saint Kitts and Nevis">Saint Kitts and  
Nevis</option>  
<option value="Saint Lucia">Saint Lucia</option>  
<option value="Saint Pierre and Miquelon">Saint Pierre and  
Miquelon</option>  
<option value="Saint Vincent and The Grenadines">Saint Vincent  
and The Grenadines</option>  
<option value="Samoa">Samoa</option>  
<option value="San Marino">San Marino</option>

<option value="Sao Tome and Principe">Sao Tome and Principe</option>

<option value="Saudi Arabia">Saudi Arabia</option>

<option value="Senegal">Senegal</option>

<option value="Serbia">Serbia</option>

<option value="Seychelles">Seychelles</option>

<option value="Sierra Leone">Sierra Leone</option>

<option value="Singapore">Singapore</option>

<option value="Slovakia">Slovakia</option>

<option value="Slovenia">Slovenia</option>

<option value="Solomon Islands">Solomon Islands</option>

<option value="Somalia">Somalia</option>

<option value="South Africa">South Africa</option>

<option value="South Georgia and The South Sandwich Islands">South Georgia and The South Sandwich Islands</option>

<option value="Spain">Spain</option>

<option value="Sri Lanka">Sri Lanka</option>

<option value="Sudan">Sudan</option>

<option value="Suriname">Suriname</option>

<option value="Svalbard and Jan Mayen">Svalbard and Jan Mayen</option>

<option value="Swaziland">Swaziland</option>

<option value="Sweden">Sweden</option>

<option value="Switzerland">Switzerland</option>

<option value="Syrian Arab Republic">Syrian Arab Republic</option>

<option value="Taiwan">Taiwan</option>

<option value="Tajikistan">Tajikistan</option>

<option value="Tanzania, United Republic of">Tanzania, United Republic of</option>

<option value="Thailand">Thailand</option>

<option value="Timor-leste">Timor-leste</option>

<option value="Togo">Togo</option>

<option value="Tokelau">Tokelau</option>

<option value="Tonga">Tonga</option>

<option value="Trinidad and Tobago">Trinidad and Tobago</option>

<option value="Tunisia">Tunisia</option>

<option value="Turkey">Turkey</option>

<option value="Turkmenistan">Turkmenistan</option>

<option value="Turks and Caicos Islands">Turks and Caicos Islands</option>

<option value="Tuvalu">Tuvalu</option>

<option value="Uganda">Uganda</option>

<option value="Ukraine">Ukraine</option>

<option value="United Arab Emirates">United Arab Emirates</option>

<option value="United Kingdom">United Kingdom</option>

<option value="United States">United States</option>

<option value="United States Minor Outlying Islands">United States Minor Outlying Islands</option>

<option value="Uruguay">Uruguay</option>

<option value="Uzbekistan">Uzbekistan</option>

<option value="Vanuatu">Vanuatu</option>

<option value="Venezuela">Venezuela</option>

<option value="Viet Nam">Viet Nam</option>

```
        <option value="Virgin Islands, British">Virgin Islands,
British</option>
        <option value="Virgin Islands, U.S.">Virgin Islands, U.S.</option>
        <option value="Wallis and Futuna">Wallis and Futuna</option>
        <option value="Western Sahara">Western Sahara</option>
        <option value="Yemen">Yemen</option>
        <option value="Zambia">Zambia</option>
        <option value="Zimbabwe">Zimbabwe</option>
    </select>
</div>

<div class="form-group">
    <input type="password" class="form-control" name="password"
placeholder="Password" id="password" required="required">
</div>

    <div class="form-group">
        <input type="password" class="form-control"
name="confirm_password" placeholder="Confirm Password"
id="confirmpassword" required="required">
    </div>

        <div class="form-group">
            <input type="submit" class="btn btn-primary btn-block" value="Sign
up">
        </div>
    </form>
</div>
</body>
</html>
```

## SIGN IN .HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
  <title>News Tracker Application</title>
```

```
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.1.3/css/bootstrap.min.css">
```

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

```
    <script src="static/sign_in.js"></script>
```

```
<style>
```

```
  .list-group {  
    margin: 10px 0;  
  }
```

```
  .list-group-item {  
    margin: 0;  
    padding: 0;  
  }
```

```
  .alert {  
    margin: 0;  
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
{% with messages = get_flashed_messages() % }
```

```
{% if messages % }
```

```
<ul class="flashes list-group">
```

```
{% for message in messages % }
```

```
<li class="list-group-item">
```

```
<div class="alert alert-success">{{ message }}</div>
```

```
</li>
```

```
{% endfor % }
```

```
</ul>
```

```
{% endif % }
```

```
{% endwith % }
```

```
<div class="register-photo">
```

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

```
<div class="image-holder"></div>
```

```
<form method="POST" action="/" name="signin" onsubmit="return  
validateform()">
```

```
<h2 class="text-center"><strong>SIGN IN</strong></h2>
```

```
<div class="form-group"><input class="form-control" type="email"  
name="email" id="email" placeholder="Email"></div>
```

```
<div class="form-group"><input class="form-control"  
type="password" name="password" id="password"  
placeholder="Password"></div>
```

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

```
        <a href="/forget_password" class="already">Forget password?  
Click here.</a>
```

```
    </div>
```

```
        <div class="form-group"><input type="submit" class="btn btn-  
primary btn-block" value="Sign in"></div><a href="/sign_up"  
class="already">Dont have an account? Sign up here.</a></form>
```

```
    </div>
```

```
</div>
```

```
    <script  
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-  
bootstrap/4.1.3/js/bootstrap.bundle.min.js"></script>
```

```
</body>
```

```
</html>
```

## OTP.HTML

```
<html>
```

```
<head><title>OTP</title>
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-  
bootstrap/4.1.3/css/bootstrap.min.css">
```

```
<style>
```

```
.card {
```

```
    width: 350px;
```

```
    padding: 10px;
```

```
    border-radius: 20px;
```

```
    background: #fff;
```

```
    border: none;
```

```
    height: 350px;
```



```
    position: relative;
}
```

```
.container {
    height: 100vh;
}
```

```
body {
    background: #eee;
}
```

```
.mobile-text {
    color: #989696b8;
    font-size: 15px;
}
```

```
.form-control {
    margin-right: 12px;
}
```

```
.form-control:focus {
    color: #495057;
    background-color: #fff;
    border-color: #ff8880;
    outline: 0;
    box-shadow: none;
}
```

```
.cursor {  
    cursor: pointer;  
}
```

```
.list-group {  
    margin: 10px 0;  
}
```

```
.list-group-item {  
    margin: 0;  
    padding: 0;  
}
```

```
.alert {  
    margin: 0;  
}
```

```
</style>
```

```
<body>
```

```
{% with messages = get_flashed_messages() %}
```

```
    {% if messages %}
```

```
    <ul class="flashes list-group">
```

```
        {% for message in messages %}
```

```
        <li class="list-group-item">
```

```

        <div class="alert alert-danger">{ { message } }</div>
    </li>
    { % endfor % }
</ul>
{ % endif % }
{ % endwith % }

<form method="POST" action="/otp">
<div class="d-flex justify-content-center align-items-center container">
    <div class="card py-5 px-3">
        <h5 class="m-0">Email verification</h5><span class="mobile-
text">Enter the code we just send to your email id <b class="text-
danger">{ { email } }</b></span>

        <div class="d-flex flex-row mt-5"><input type="text" class="form-
control" autofocus="" name="box1"><input type="text" class="form-control"
name="box2"><input type="text" class="form-control" name="box3"><input
type="text" class="form-control" name="box4"></div>

        <br><div class="form-group">

            <input type="submit" class="btn btn-primary btn-block" value="Verify
otp">

        </div>

        <div><span class="d-block mobile-text"><center>Don't receive the
code?</center></span><br><center><span class="font-weight-bold text-danger
cursor"><a href="/resend_otp">Resend</a></span></center></div>

</div>

</div>

```

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/js/bootstrap.bundle.min.js"></script>

</form>

</body>

</html>
```

## HOME.HTML

```
<!doctype html>

<html>

<head><title>Home</title>

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

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.j
s" integrity="sha384-
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOm
Mi466C8" crossorigin="anonymous"></script>

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

<style>

html,

body {

height: 100%;

margin: 0;

background-image: linear-gradient(90deg, #020024 0%, #0920a9
50%, #005fff 100%);

}
```

```
.box {  
  display: flex;  
  flex-flow: column;  
  height: 100%;  
}  
.box .row {  
  border: 1px dotted #0313fc;  
}  
.box .row.header {  
  flex: 0 1 auto;  
}  
.box .row.content {  
  flex: 1 1 auto;  
}  
.box .row.footer {  
  flex: 0 1 40px;  
}
```

```
h2 {  
  text-decoration:underline;  
}
```

```
</style>
```

```
<script>
```

```

window.watsonAssistantChatOptions = {
  integrationID: "", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "", // The ID of your service instance.
  onLoad: function(instance) { instance.render(); }
};

setTimeout(function(){
  const t=document.createElement('script');

  t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";

  document.head.appendChild(t);

});
</script>

</head>

<body style= "background-color:GhostWhite">

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCO
mLASjC" crossorigin="anonymous">

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.j
s" integrity="sha384-
IQsoLX15PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4
GN1R8p" crossorigin="anonymous"></script>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">

```



```

    <form method="POST" action="/search" class="d-flex" role="search">
        <input class="form-control me-2" type="search" placeholder="Search"
        aria-label="Search" name="search">

        <input type="submit" class="btn btn-outline-success" value="Search">
    </form>
</div>
</div>
</nav>
<hr style="color:white ; height:5px;">

```

```

<div class="box" style="float:right; width:75%; padding-left:20px; padding-
right:20px;">

```

```

    <h2>{{ fav1 }}</h2>

    <br>

    {% if c1_title and c1_url %}

    {% for i in range(c1_length) %}

        <a href ={{ c1_url[i] }} style="width:100%;height:200px;float:left
;background:Black;text-decoration:none; font-size:25px; color:white;
display:flex; padding-left:20px;"><img src={{ c1_urltoimage[i] }}
style="width:100px;height:100px; float:left; ">{{ c1_title[i] }}</a>

        <br>

    {% endfor %}

    {% endif %}

    <br>

    <br>

    <h2>{{ fav2 }}</h2>

```



```

<br>

{% if c2_title and c2_url %}

    {% for i in range(c2_length) %}

        <a href={{c2_url[i]}} style="width:100%;height:200px;float:left
;background:Black;text-decoration:none; font-size:25px; color:white;
display:flex; padding-left:20px;"><img src={{c2_urltoimage[i]}}
style="width:100px;height:100px; float:left; ">{{c2_title[i]}}</a>

        <br>

    {% endfor %}

{% endif %}


<br>

<br>

<h2>{{fav3}}</h2>

<br>

{% if c3_title and c3_url %}

    {% for i in range(c3_length) %}

        <a href={{c3_url[i]}} style="width:100%;height:200px;float:left
;background:Black;text-decoration:none; font-size:25px; color:white;
display:flex; padding-left:20px;"><img src={{c3_urltoimage[i]}}
style="width:100px;height:100px; float:left; ">{{c3_title[i]}}</a>

        <br>

    {% endfor %}

{% endif %}

</div>


{% if news_title and url %}

    {% for i in range(main_length) %}

```

```

<div class="card" style="width: 18rem; background-color:Black;" >
  <img src={{ urltoimage[i] }} class="card-img-top" alt="..." >
  <div class="card-body" >
    <h5 class="card-title" style="color:white">Breaking news</h5>
    <a href={{ url[i] }} class="card-text"
style="color:white">{{ news_title[i] }}</a>
  </div>
</div><br>
{% endfor %}
{% endif %}
</body>

```

## PROFILE.HTML

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="static/pstyle.css">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/d
AiS6JXm" crossorigin="anonymous">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.14.0/css/all.min.css" integrity="sha512-
1PKOgIY59xJ8Co8+NE6FZ+LOAZKjy+KY8iq0G4B3CyeY6wYHN3yt9PW0
XpSriVlkMXe40PTKnXrLnZ9+fkDaog==" crossorigin="anonymous" />
<script src="static/edit.js"></script>

<style>
html,

```

```
body {  
  height: 100%;  
  margin: 0;  
  background-image: linear-gradient(90deg, #020024 0%, #0920a9  
50%, #005fff 100%);  
}  
.box {  
  display: flex;  
  flex-flow: column;  
  height: 100%;  
}  
.box .row {  
  border: 1px dotted #0313fc;  
}  
.box .row.header {  
  flex: 0 1 auto;  
}  
.box .row.content {  
  flex: 1 1 auto;  
}  
.box .row.footer {  
  flex: 0 1 40px;  
}  
  
h2 {  
  text-decoration: underline;  
}
```

```
</style>

</head>

<body style= "background-color:GhostWhite">

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCO
mLASjC" crossorigin="anonymous">

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.j
s" integrity="sha384-
IQsoLX15PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4
GN1R8p" crossorigin="anonymous"></script>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link" href="/home">Home</a>
        </li>
        <li class="nav-item">
```



```
<form method="POST" action="/profile" name="profile" onsubmit="return validateform()">
```

```
<div class="container-xl px-4 mt-4">
```

```
  <hr class="mt-0 mb-4">
```

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

```
    <div class="col-xl-4">
```

```
      <!-- Profile picture card-->
```

```
      <div class="card mb-4 mb-xl-0">
```

```
        <div class="card-header">Favourites:</div>
```

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

```
          <!-- Profile picture image-->
```

```
      <div class="mb-3">
```

```
        <label class="small mb-1" for="choice1">Choice  
1: { { account['FAV1'] } }</label>
```

```
        <select class="form-control" name="choice1" id="choice1" >
```

```
          <option value="0" selected>Change</option>
```

```
          <option value="Business">Business</option>
```

```
          <option value="Entertainment">Entertainment</option>
```

```
          <option value="General">General</option>
```

```
          <option value="Health">Health</option>
```

```
          <option value="Science">Science</option>
```

```
          <option value="Technology">Technology</option>
```

```
        </select>
```

```
      </div>
```

```
<div class="mb-3">
```

```
  <label class="small mb-1" for="choice2">Choice  
2:{{ account['FAV2']}}</label>
```

```
  <select class="form-control" name="choice2" id="choice2" >  
    <option value="0" selected>Change</option>  
    <option value="Business">Business</option>  
    <option value="Entertainment">Entertainment</option>  
    <option value="General">General</option>  
    <option value="Health">Health</option>  
    <option value="Science">Science</option>  
    <option value="Technology">Technology</option>  
  </select>
```

```
</div>
```

```
<div class="mb-3">
```

```
  <label class="small mb-1" for="choice3">Choice  
3:{{ account['FAV3']}}</label>
```

```
  <select class="form-control" name="choice3" id="choice3">  
    <option value="0" selected>Change</option>  
    <option value="Business">Business</option>  
    <option value="Entertainment">Entertainment</option>  
    <option value="General">General</option>  
    <option value="Health">Health</option>  
    <option value="Science">Science</option>  
    <option value="Technology">Technology</option>  
  </select>
```

```
</div>
```

```

        </div>
    </div>
</div>
<div class="col-xl-8">
    <!-- Account details card-->
    <div class="card mb-4">
        <div class="card-header">Account Details</div>
        <div class="card-body">

            <!-- Form Group (username)-->
            <div class="mb-3">
                <label class="small mb-1 "
for="inputUsername">Username</label>

                <input class="form-control" id="inputUsername" type="text"
placeholder="Enter your username" value={ { account['NAME']} }
name="name">

            </div>

            <!-- Form Row      -->
            <div class="row gx-3 mb-3">
                <!-- Form Group (organization name)-->
                <div class="col-md-6">
                    <label class="small mb-1 "
for="inputOrgName">Password</label>

```



```
        <input class="form-control" id="inputOrgName"
type="text" placeholder="Password" value={{ account['PASSWORD'] }}
name="password">

    </div>

    <!-- Form Group (location)-->

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

        <label class="small mb-1"
for="inputLocation">Country</label>

        <input class="form-control" id="inputLocation" type="text"
placeholder="Enter your location" value={{ account['COUNTRY'] }}
name="country">

    </div>

</div>

<!-- Form Group (email address)-->

<div class="mb-3">

    <label class="small mb-1" for="inputEmailAddress">Email
address</label>

    <input class="form-control" id="inputEmailAddress"
type="email" placeholder="Enter your email address"
value={{ account['EMAIL'] }} name="email" disabled="disabled">

</div>

<!-- Form Row-->

<div class="row gx-3 mb-3">

    <!-- Form Group (phone number)-->

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

        <label class="small mb-1" for="inputPhone">Phone
number</label>

        <input class="form-control" id="inputPhone" type="tel"
placeholder="Enter your phone number" value={{ account['PHONE'] }}
name="phone">

    </div>
```

```
        <!-- Save changes button-->
        <div class="mb-3">
        <div class="mb-3">
            <a href="/home" class="btn btn-primary">Back</a>
            <input class="btn btn-primary" type="submit" value="Save
changes"></div>
        </div>
    </div>
```

```

</div>
</div>
</div>
</div>
</div>
</form>
</body>
</html>
```

## **CHOICE.HTML**

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
/*the container must be positioned relative:*/
```

```
.custom-select {  
  position: relative;  
  font-family: Arial;  
}
```

```
.styled {  
  border: 0;  
  line-height: 2.5;  
  padding: 0 20px;  
  font-size: 1rem;  
  text-align: center;  
  color: #fff;  
  text-shadow: 1px 1px 1px #000;  
  border-radius: 10px;  
  background-color: rgba(0,0,255,0.3);  
  background-image: linear-gradient(to top left,  
                                   rgba(0, 0, 0, .2),  
                                   rgba(0, 0, 0, .2) 30%,  
                                   rgba(0, 0, 0, 0));  
  box-shadow: inset 2px 2px 3px rgba(255, 255, 255, .6),  
             inset -2px -2px 3px rgba(0, 0, 0, .6);  
}
```

```
.styled:hover {  
  background-color: rgba(255, 0, 0, 1);  
}
```

```
.styled:active {  
    box-shadow: inset -2px -2px 3px rgba(255, 255, 255, .6),  
                inset 2px 2px 3px rgba(0, 0, 0, .6);  
}
```

```
.custom-select select {  
    display: none; /*hide original SELECT element:*/  
  
}
```

```
.select-selected {  
    background-color: DodgerBlue;  
}
```

```
/*style the arrow inside the select element:*/  
.select-selected:after {  
    position: absolute;  
    content: "";  
    top: 14px;  
    right: 10px;  
    width: 0;  
    height: 0;  
    border: 6px solid transparent;
```

```
border-color: #fff transparent transparent transparent;
}
```

```
/*point the arrow upwards when the select box is open (active):*/
.select-selected.select-arrow-active:after {
border-color: transparent transparent #fff transparent;
top: 7px;
}
```

```
/*style the items (options), including the selected item:*/
.select-items div,.select-selected {
color: #ffffff;
padding: 8px 16px;
border: 1px solid transparent;
border-color: transparent transparent rgba(0, 0, 0, 0.1) transparent;
cursor: pointer;
user-select: none;
}
```

```
/*style items (options):*/
.select-items {
position: absolute;
background-color: DodgerBlue;
top: 100%;
left: 0;
right: 0;
z-index: 99;
```

```
}
```

```
/*hide the items when the select box is closed:*/
```

```
.select-hide {  
    display: none;  
}
```

```
.select-items div:hover, .same-as-selected {  
    background-color: rgba(0, 0, 0, 0.1);  
}
```

```
h2 {text-align : center};
```

```
.list-group {  
    margin: 10px 0;  
}
```

```
.list-group-item {  
    margin: 0;  
    padding: 0;  
}
```

```
.alert {  
    margin: 0;  
}
```

```
</style>
```

```
</head>
```

```
<body>
{ % with messages = get_flashed_messages() % }
    { % if messages % }
        <ul class="flashes list-group">
            { % for message in messages % }
                <li class="list-group-item">
                    <div class="alert alert-success">{ { message } }</div>
                </li>
            { % endfor % }
        </ul>
    { % endif % }
{ % endwith % }
```

```
<h2>Select your favourite choice</h2>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<!--surround the select box with a "custom-select" DIV element. Remember to
set the width:-->
```

```
<center>
```

```
    <form method="POST" action="/choice">
```

```
<div class="custom-select" class="styled" id="right" style="width:300px;" >
```

```
    <select name="choice1" id="choice1">
```

```
        <option value="0">Select:</option>
```

```
<option value="Business">Business</option>
<option value="Entertainment">Entertainment</option>
<option value="General">General</option>
<option value="Health">Health</option>
<option value="Science">Science</option>
<option value="Technology">Technology</option>
</select>
</div>
<br>
<br>
<br>
<br>
```

```
<div class="custom-select" id="left" style="width:300px;" >
  <select name="choice2" id="choice2">
    <option value="0">Select:</option>
    <option value="Business">Business</option>
    <option value="Entertainment">Entertainment</option>
    <option value="General">General</option>
    <option value="Health">Health</option>
    <option value="Science">Science</option>
    <option value="Technology">Technology</option>
  </select>
</div>

<br>
```



<br>

<br>

<br>

<div class="custom-select" id="center" style="width:300px;" >

<select name="choice3" id="choice3">

<option value="0">Select:</option>

<option value="Business">Business</option>

<option value="Entertainment">Entertainment</option>

<option value="General">General</option>

<option value="Health">Health</option>

<option value="Science">Science</option>

<option value="Technology">Technology</option>

</select>

</div>

<br>

<br>

<br>

<br>

<input class="favorite styled"

type="submit"

value="Add to favorites">

</form>

</center>

```
<script>
var x, i, j, l, ll, selElmnt, a, b, c;
/*look for any elements with the class "custom-select":*/
x = document.getElementsByClassName("custom-select");
l = x.length;
for (i = 0; i < l; i++) {
    selElmnt = x[i].getElementsByTagName("select")[0];
    ll = selElmnt.length;
    /*for each element, create a new DIV that will act as the selected item:*/
    a = document.createElement("DIV");
    a.setAttribute("class", "select-selected");
    a.innerHTML = selElmnt.options[selElmnt.selectedIndex].innerHTML;
    x[i].appendChild(a);
    /*for each element, create a new DIV that will contain the option list:*/
    b = document.createElement("DIV");
    b.setAttribute("class", "select-items select-hide");
    for (j = 1; j < ll; j++) {
        /*for each option in the original select element,
        create a new DIV that will act as an option item:*/
        c = document.createElement("DIV");
        c.innerHTML = selElmnt.options[j].innerHTML;
        c.addEventListener("click", function(e) {
            /*when an item is clicked, update the original select box,
            and the selected item:*/
            var y, i, k, s, h, sl, yl;
```

```
s = this.parentNode.parentNode.getElementsByTagName("select")[0];
sl = s.length;
h = this.parentNode.previousSibling;
for (i = 0; i < sl; i++) {
    if (s.options[i].innerHTML == this.innerHTML) {
        s.selectedIndex = i;
        h.innerHTML = this.innerHTML;
        y = this.parentNode.getElementsByTagName("same-as-selected");
        yl = y.length;
        for (k = 0; k < yl; k++) {
            y[k].removeAttribute("class");
        }
        this.setAttribute("class", "same-as-selected");
        break;
    }
}
h.click();
});
b.appendChild(c);
}
x[i].appendChild(b);
a.addEventListener("click", function(e) {
    /*when the select box is clicked, close any other select boxes,
    and open/close the current select box:*/
    e.stopPropagation();
    closeAllSelect(this);
    this.nextSibling.classList.toggle("select-hide");
```

```
        this.classList.toggle("select-arrow-active");
    });
}
function closeAllSelect(elmnt) {
    /*a function that will close all select boxes in the document,
    except the current select box:*/
    var x, y, i, xl, yl, arrNo = [];
    x = document.getElementsByClassName("select-items");
    y = document.getElementsByClassName("select-selected");
    xl = x.length;
    yl = y.length;
    for (i = 0; i < yl; i++) {
        if (elmnt == y[i]) {
            arrNo.push(i)
        } else {
            y[i].classList.remove("select-arrow-active");
        }
    }
    for (i = 0; i < xl; i++) {
        if (arrNo.indexOf(i)) {
            x[i].classList.add("select-hide");
        }
    }
}
/*if the user clicks anywhere outside the select box,
then close all select boxes:*/
document.addEventListener("click", closeAllSelect);
```

</script>

</body>

</html>

## **DUMMY.HTML**

<!doctype html>

<title>My Application</title>

{% with messages = get\_flashed\_messages() % }

{% if messages % }

<ul class=flashes>

{% for message in messages % }

<li>{{ message }}</li>

{% endfor % }

</ul>

{% endif % }

{% endwith % }

## **SEARCH\_NEWS.HTML**

<!doctype html>

<html>

<head><title>Home</title>

<link

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

rel="stylesheet" integrity="sha384-

iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.j

s" integrity="sha384-

u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8" crossorigin="anonymous"></script>

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

```
<style>
```

```
  html,
```

```
  body {
```

```
    height: 100%;
```

```
    margin: 0;
```

```
  }
```

```
  .box {
```

```
    display: flex;
```

```
    flex-flow: column;
```

```
    height: 100%;
```

```
  }
```

```
  .box .row {
```

```
    border: 1px dotted #0313fc;
```

```
  }
```

```
  .box .row.header {
```

```
    flex: 0 1 auto;
```

```
  }
```

```
  .box .row.content {
```

```
    flex: 1 1 auto;
```

```
  }
```

```
  .box .row.footer {
```

```
    flex: 0 1 40px;
```

```
  }
```

```
  h2 {
```

```
text-decoration:underline;
}
```

```
</style>
```

```
<script>
```

```
window.watsonAssistantChatOptions = {
  integrationID: "", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "", // The ID of your service instance.
  onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
  const t=document.createElement('script');
  t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
  document.head.appendChild(t);
});
```

```
</script>
```

```
</head>
```

```
<body style= "background-color:GhostWhite">
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
```

EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<script  
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLX15PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">



<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

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

</li>

<li class="nav-item">

<a class="nav-link" href="/profile">Profile</a>

</li>

<li class="nav-item">

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

</li>

<li class="nav-item">





```
alt="alternatetext" style="width:100px;height:100px; float:left;
">{{ search_title[i] }}</a>
```

```
<br>
```

```
{% endfor %}
```

```
{% endif %}
```

```
</div>
```

```
</body>
```

```
</html>
```

## **FORGET\_PASSWORD.HTML**

```
<!DOCTYPE html>
```

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

```
<head>
```

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

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

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

```
<title>Forgot Password</title>
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
<script src="static/forget_pwd.js"></script>
```

```
<style>
```

```
.list-group {
```

```
margin: 10px 0;
```

```
}
```

```
.list-group-item {  
    margin: 0;  
    padding: 0;  
}
```

```
.alert {  
    margin: 0;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
{% with messages = get_flashed_messages() %}
```

```
{% if messages %}
```

```
<ul class="flashes list-group">
```

```
    {% for message in messages %}
```

```
    <li class="list-group-item">
```

```
        <div class="alert alert-danger">{{ message }}</div>
```

```
    </li>
```

```
    {% endfor %}
```

```
</ul>
```

```
{% endif %}
```

```
{% endwith % }
```

```
<div class="container d-flex flex-column">
  <div class="row align-items-center justify-content-center
    min-vh-100">
    <div class="col-12 col-md-8 col-lg-4">
      <div class="card shadow-sm">
        <div class="card-body">
          <div class="mb-4">
            <h5>Forgot Password?</h5>
            <p class="mb-2">Enter your registered email ID to send your
password
            </p>
          </div>
          <form method="POST" action="/forget_password" name="forgetpwd"
onsubmit="return validateform()">
            <div class="mb-3">
              <label for="email" class="form-label">Email</label>
              <input type="email" id="email" class="form-control" name="email"
placeholder="Enter Your Email" id="email"
              required="">
            </div>
            <div class="mb-3 d-grid">
              <input type="submit" class="btn btn-primary value="Send
Password>
              </input>
            </div>
            <span>Don't have an account? <a href="/sign_up">sign
up</a></span>
          </form>
```

```
        </div>
    </div>
</div>
</div>
</div>
</body>
</html>
```

## **EDIT.JS**

```
function validateform(){
var name=document.forms["profile"]["name"].value;
var email=document.forms["profile"]["email"].value;
var password=document.forms["profile"]["password"].value;
var country=document.forms["profile"]["country"].value;
var phone=document.forms["profile"]["phone"].value;
```

```
if (name==null || name==""){
    alert("Name can't be empty");
    return false;
}
```

```
else if (email==null || email==""){
    alert("Email can't be empty");
    return false;
}
```

```
else if (password==null || password==""){  
    alert("Pasword can't be empty");  
    return false;  
}
```

```
else if(!(/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/).test(email)))  
  
    alert("You have entered an invalid email address!");  
    return false  
}
```

```
else if(password.length<6){  
    alert("Password must be at least 6 characters long.");  
    return false;  
}
```

```
if (country==null || country==""){  
    alert("Country field is empty");  
    return false;  
}
```

```
else if (phone==null || phone==""){
```

```
    alert("Phone No can't be empty");  
    return false;  
}
```

```
else if ((phone.length)!=10){  
    alert("Enter a valid Phone No");  
    return false;  
}
```

```
confirm("Are you sure you want to save changes")  
  
}
```

### **FORGET\_PWD.JS**

```
function validateform(){  
var email=document.forms["forgetpwd"]["email"].value;
```

```
if (email==null || email==""){  
    alert("Email can't be empty");  
    return false;  
}
```

```
else if (!(/^w+([\\.-]?w+)*@w+([\\.-]?w+)*\\.w{2,3}+$/).test(email)){
```

```
    alert("You have entered an invalid email address!");  
    return false  
}
```

```
}
```

### **SIGN\_IN.HTML**

```
function validateform(){
```

```
var email=document.forms["signin"]["email"].value;
```

```
var pass=document.forms["signin"]["password"].value;
```

```
if (email==null || email==""){
```

```
    alert("Email can't be empty");
```

```
    return false;
```

```
}
```

```
else if (pass==null || pass==""){
```

```
    alert("Password can't be empty");
```

```
    return false;
```

```
}
```

```
else if(!(/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/).test(email))){
```

```
    alert("You have entered an invalid email address!");
```

```
    return false
```

```
}
```

```
}
```

### **SIGN\_UP.HTML**

```
function validateform(){
```

```
var name=document.forms["signup"]["username"].value;
```

```
var email=document.forms["signup"]["email"].value;
```



```
var password=document.forms["signup"]["password"].value;
var confirmpwd=document.forms["signup"]["confirmpassword"].value;
var country=document.forms["signup"]["country"].value;
var phone=document.forms["signup"]["phone"].value;
if (name==null || name==""){
    alert("Name can't be empty");
    return false;
}

else if (email==null || email==""){
    alert("Email can't be empty");
    return false;
}

else if (password==null || password==""){
    alert("Pasword can't be empty");
    return false;
}

else if (confirmpwd==null || confirmpwd==""){
    alert("Confirm Pasword can't be empty");
    return false;
}

else if(!(/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/).test(email)){

    alert("You have entered an invalid email address!");
    return false
}
```

```
else if(password.length<6){
    alert("Password must be at least 6 characters long.");
    return false;
}
else if(password!=confirmpwd){
    alert("Password mismatch");
    return false
}
else if (country=="select a country"){
    alert("Select a country");
    return false;
}
else if (phone==null || phone==""){
    alert("Phone No can't be empty");
    return false;
}

else if ((phone.length)!=10){
    alert("Enter a valid Phone No");
    return false;
}
}
```

## **STYLE.CSS**

```
.register-photo {
    background:#f1f7fc;
    padding:80px 0;
```

```
}
```

```
.register-photo .image-holder {  
  display:table-cell;  
  width:auto;  
  background:url(/static/newsicon.jpg);  
  background-size:cover;  
}
```

```
.register-photo .form-container {  
  display:table;  
  max-width:900px;  
  width:90%;  
  margin:0 auto;  
  box-shadow:1px 1px 5px rgba(7,15,258,0.72);  
}
```

```
.register-photo form {  
  display:table-cell;  
  width:400px;  
  background-color:#ffffff;  
  padding:40px 60px;  
  color:#505e6c;  
}
```

```
@media (max-width:991px) {  
  .register-photo form {
```

```
padding:40px;  
}  
}
```

```
.register-photo form h2 {  
font-size:24px;  
line-height:1.5;  
margin-bottom:30px;  
}
```

```
.register-photo form .form-control {  
background:#f7f9fc;  
border:none;  
border-bottom:1px solid #dfe7f1;  
border-radius:0;  
box-shadow:none;  
outline:none;  
color:inherit;  
text-indent:6px;  
height:40px;  
}
```

```
.register-photo form .form-check {  
font-size:13px;  
line-height:20px;  
}
```

```
.register-photo form .btn-primary {  
  background:#f4476b;  
  border:none;  
  border-radius:4px;  
  padding:1 1px;  
  box-shadow:none;  
  margin-top:35px;  
  text-shadow:none;  
  outline:none !important;  
}
```

```
.register-photo form .btn-primary:hover, .register-photo form .btn-  
primary:active {  
  background:#eb3b60;  
}
```

```
.register-photo form .btn-primary:active {  
  transform:translateY(1px);  
}
```

```
.register-photo form .already {  
  display:block;  
  text-align:center;  
  font-size:12px;  
  color:#6f7a85;  
  opacity:0.9;  
  text-decoration:none;  
}
```

```
.left_container{  
  float:left;  
  width:30%  
}
```

```
.right_container{  
  float:right;  
  width:70%  
}
```

### **PSTYLE.CSS**

```
body{ margin-top:20px;  
background-color:#f2f6fc;  
color:#69707a;  
}
```

```
.img-account-profile {  
  height: 10rem;  
}
```

```
.rounded-circle {  
  border-radius: 50% !important;  
}
```

```
.card {  
  box-shadow: 0 0.15rem 1.75rem 0 rgb(33 40 50 / 15%);  
}
```

```
.card .card-header {  
  font-weight: 500;  
}
```

```
.card-header:first-child {
```

```
border-radius: 0.35rem 0.35rem 0 0;
}

.card-header {
padding: 1rem 1.35rem;
margin-bottom: 0;
background-color: rgba(7,15,258,0.72);
border-bottom: 1px solid rgba(7,15,258,0.72);
}

.form-control, .dataTable-input {
display: block;
width: 100%;
padding: 0.875rem 1.125rem;
font-size: 0.875rem;
font-weight: 400;
line-height: 1;
color: #69707a;
background-color: #fff;
background-clip: padding-box;
border: 1px solid #c5ccd6;
-webkit-appearance: none;
-moz-appearance: none;
appearance: none;
border-radius: 0.35rem;
transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;
}

.nav-borders .nav-link.active {
```

```

color: #0061f2;
border-bottom-color: #0061f2;
}
.nav-borders .nav-link {
color: #69707a;
border-bottom-width: 0.125rem;
border-bottom-style: solid;
border-bottom-color: transparent;
padding-top: 0.5rem;
padding-bottom: 0.5rem;
padding-left: 0;
padding-right: 0;
margin-left: 1rem;
margin-right: 1rem;
}

```

## 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 sign in option	Can visible	Yes visible	pass	Successful	-	Prabu
Sign up	Functional	Login page	Verify user has the option to sign up	Can visible	Yes visible	pass	Successful	-	Syed Ibrahim
Forget password	Functional	Login page	Verify user has the option to forgot password	Yes, the option is available	Option is available	pass	Successful	-	Nithish Kumar
Fetch news	Functional	Home page	Verify user can get the news	News will be feed to the app	404 error	Fail	Unsuccessful	App integration problem	Vishal
Type of news available in the fetch news page	Functional	Fetch news	Types of news available	Weather, Sport, Economy	Hover buttons Will be shown	Yes	Successful	-	Vishal, Nithish Kumar

The development and execution of the cloud-based news tracking application is complete.



## 9.1 Performance Metrics

### *NFT- Risk Assessment*

S.no	Project Name	Scope/ feature	Functional changes	Hardware changes	Software changes	Impact of Downtime	Load/Volume Changes	Risk Score	Justification
1.	News Tracker Application	Tracks News Based on the users Specification.	No	No Changes	Low	>10sec	>5 to 10%	5%	Tracks and Shows the news based on user recommendation.

### *NFT- Detailed Test Plan*

S.no	Project Overview	NFT Test approach	Assumptions/ Dependencies/Risks	Approvals/ Sign Off
1.	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.	Load Testing	No Risks	Vishal, Nithish Kumar
2.	Multiple user login and accessing the application.	Stress Testing	>20 Users	Prabu, Syed Ibrahim

### *End of Test Report*

S.no	Project Overview	NFT Test approach	NFR- Met	Test Outcome	Go/ No-Go decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/ Sign Off
1.	How long the time period of the application.	Endurance Testing	Yes	Moderate >8 hours	Go	Runs >8 hours in a single run of the application.	Closed	Vishal, Prabu
2.	Increase the user load for a short amount of time.	Spike Testing	Yes	Low <1 hour	Go	Runs <1 hours in a single run of the application.	Open	Nithish Kumar, Syed Ibrahim

## 10.CONCLUSION

We investigated the viability of identifying interaction patterns in reading news and assessed three adaptive interface designs for various news reader categories. We demonstrate how a specific user can be identified as one of three types based on their interaction record. The online survey revealed several different and well-defined reader types. Different news reader types require various user interfaces, according to the examination of the three variation interfaces. We have shown how to track users' news reading habits and determine their news reader type from it. In order to be able to demonstrate a complete adaptive mobile news framework that provides automatic personalization of news apps, we will continue to investigate the design of adaptive interfaces in the future.

## 11.ADVANTAGES&DISADVANTAGES

### ADVANTAGES:

- News is already categorised in this app.

- Better user experience; easily portable; easier access.
- Apps might help you turn website visitors into devoted readers.
- Minute-by-minute news updates.
- This app enables you to rapidly receive local news updates.
- To research and identify hot topics and news.
- News feeds tailored to your interests for you.

### **DISADVANTAGES:**

- Some apps need a user to subscribe to a premium version.
- When advertisements appear, they disrupt users.
- The news occasionally provides succinct information.
- The prevalence of false and ambiguous news might mislead users and cause confusion.
- Readers may be misled by fake news.

### **12.FUTURE SCOPE**

In order to be able to present a complete adaptive mobile news framework that provides automatic personalization of news apps, we will continue to investigate the design of adaptive interfaces in the future.

### **13.APPENDIX**

#### **MAIN.PY**

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

```
import ibm_db
```

```
from headlines import news_headlines
```

```
from headlines import news_category
```

```
from mail import *
```

```
from key_search import search_by_key
```

```
conn = ibm_db.connect(
```

```
    "DATABASE=bludb;HOSTNAME=2f3279a5-73d1-4859-88f0-  
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=307  
56;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=fp  
x81088;PWD=tdzPJpeAcG7bPkGW",
```

", ")

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

```
app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'
```

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

```
def sign_in():
```

```
    all_users=[]
```

```
    if request.method=='POST':
```

```
        user_email=request.form['email']
```

```
        user_password=request.form['password']
```

```
        session['email'] = request.form['email']
```

```
        print(user_email,user_password)
```

```
        sql = "SELECT * FROM user"
```

```
        stmt = ibm_db.exec_immediate(conn, sql)
```

```
        users = ibm_db.fetch_both(stmt)
```

```
        while users != False:
```

```
            # print ("The Name is : ", dictionary)
```

```
            all_users.append(users)
```

```
            users = ibm_db.fetch_both(stmt)
```

```
        for user in all_users:
```

```
            print(user)
```

```
            if user_email == user['EMAIL']:
```

```
                if user_password == user['PASSWORD']:
```

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

```
flash("Account doesn't exist")
return render_template('sign_in.html')
```

```
else:
```

```
    if 'email' in session:
        return redirect('/home')
    return render_template('sign_in.html')
```

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

```
def sign_up():
```

```
    all_users = []
```

```
    if request.method == 'POST':
```

```
        user_name = request.form['name']
```

```
        user_email = request.form['email']
```

```
        user_country = request.form['country']
```

```
        user_phone = request.form['phone']
```

```
        user_password = request.form['password']
```

```
        user_confirm_password = request.form['confirm_password']
```

```
print(user_name,user_email,user_country,user_phone,user_password,user_confirm_password)
```

```
    sql = "SELECT * FROM user"
```

```
    stmt = ibm_db.exec_immediate(conn, sql)
```

```
    users = ibm_db.fetch_both(stmt)
```

```
    while users != False:
```

```

        # print ("The Name is : ", dictionary)
        all_users.append(users)
        users = ibm_db.fetch_both(stmt)
    for user in all_users:
        if user_email == user['EMAIL']:
            session.pop('email', None)
            flash("Account already exist")
            return redirect("/")

    session['email'] = request.form['email']
    session['name'] = request.form['name']
    session['country'] = request.form['country']
    session['phone'] = request.form['phone']
    session['password'] = request.form['password']
    return redirect('/otp')

else:
    return render_template('sign_up.html')

@app.route('/forget_password',methods=['POST','GET'])
def forget_password():
    if request.method == 'POST':
        user_email = request.form['email']
        print(user_email)
        sql = "SELECT PASSWORD FROM user WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, user_email)

```

```

    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    if account:
        print(account['PASSWORD'])
        pwd="<strong>PASSWORD:<strong>" + account['PASSWORD']
        send_mail(user_email, "Your password has been received",pwd,None)
        session.pop('email', None)
        flash('Your password has been sent')
        return redirect("/")
    else:
        flash("Account doesn't exist")
        return redirect('/forget_password')

else:
    return render_template('forget_password.html')

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

    if request.method == 'POST':
        num_1 = request.form['box1']
        num_2 = request.form['box2']
        num_3 = request.form['box3']
        num_4 = request.form['box4']
        my_otp=num_1+num_2+num_3+num_4

        if my_otp==session['otp']:

```

```
print(my_otp)
print(session['otp'])
session.pop('otp', None)
flash("OTP Verified")
return redirect('/choice')
```

else:

```
print(my_otp)
print(session['otp'])
session.pop('otp', None)
flash("OTP is wrong")
return redirect('/otp')
```

else:

```
session['otp'] = str(generate_number())
send_mail(session['email'], "Otp from news tracker", session['otp'], "yes")
email=session['email']
return render_template('otp.html',email=email)
```

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

```
def resend_otp():
```

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

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

```
def home():
```

```
    sql = "SELECT * FROM user WHERE email =?"
```

```
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, session['email'])
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
country=account['COUNTRY']
fav1=account['FAV1']
fav2 = account['FAV2']
fav3 = account['FAV3']
```

```
title, url, urltoimage = news_headlines(country)
c1_title,c1_url,c1_urltoimage=news_category(country,fav1)
c2_title, c2_url, c2_urltoimage = news_category(country,fav2)
c3_title, c3_url, c3_urltoimage = news_category(country, fav3)
main_length=len(title)
c1_length=len(c1_title)
c2_length=len(c2_title)
c3_length = len(c3_title)
return
render_template('home.html',news_title=title,url=url,urltoimage=urltoimage,
                c1_title=c1_title,c1_url=c1_url,c1_urltoimage=c1_urltoimage,
                c2_title=c2_title, c2_url=c2_url,
c2_urltoimage=c2_urltoimage,
                c3_title=c3_title, c3_url=c3_url,
c3_urltoimage=c3_urltoimage,
                main_length=main_length,c1_length=18,c2_length=18,
                c3_length=18,fav1=fav1,fav2=fav2,fav3=fav3)
```



)

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

```
def search():
```

```
    if request.method == 'POST':
```

```
        keyword = request.form['search']
```

```
        search_title,search_url,search_urltoimage=search_by_key(keyword)
```

```
        search_length=len(search_title)
```

```
        print(search_url)
```

```
        return render_template("search_news.html",search_length=search_length,  
                                search_title=search_title,
```

```
                                search_url=search_url,search_urltoimage=search_urltoimage)
```

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

```
def profile():
```

```
    if request.method=='POST':
```

```
        user_email=session['email']
```

```
        sql = "SELECT * FROM user WHERE email =?"
```

```
        stmt = ibm_db.prepare(conn, sql)
```

```
        ibm_db.bind_param(stmt, 1, user_email)
```

```
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
```

```
present_choice1 = account['FAV1']
present_choice2 = account['FAV2']
present_choice3 = account['FAV3']
```

```
sql="DELETE FROM user WHERE email =?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt, 1, user_email)
ibm_db.execute(stmt)
```

```
user_name = request.form['name']
user_email = session['email']
user_country = request.form['country']
user_phone = request.form['phone']
user_password = request.form['password']
choice1 = request.form['choice1']
choice2 = request.form['choice2']
choice3 = request.form['choice3']
```

```
if choice1=="0":
    choice1=present_choice1
```

```
if choice2=="0":
    choice2=present_choice2
```

```
if choice3=="0":  
    choice3=present_choice3
```

```
print(user_name,user_email,user_password,user_phone,choice1,choice2,choice  
3)
```

```
print(present_choice1,present_choice2,present_choice3)
```

```
insert_sql = "INSERT INTO user VALUES (?,?,,?,?,,?,?)"
```

```
prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
ibm_db.bind_param(prepare_stmt, 1, user_email)
```

```
ibm_db.bind_param(prepare_stmt, 2, user_name)
```

```
ibm_db.bind_param(prepare_stmt, 3, user_phone)
```

```
ibm_db.bind_param(prepare_stmt, 4, user_country)
```

```
ibm_db.bind_param(prepare_stmt, 5, user_password)
```

```
ibm_db.bind_param(prepare_stmt, 6, choice1)
```

```
ibm_db.bind_param(prepare_stmt, 7, choice2)
```

```
ibm_db.bind_param(prepare_stmt, 8, choice3)
```

```
ibm_db.execute(prepare_stmt)
```

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

```
user_email = session['email']
```

```
print(user_email)

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

stmt = ibm_db.prepare(conn, sql)

ibm_db.bind_param(stmt, 1, user_email)

ibm_db.execute(stmt)

account = ibm_db.fetch_assoc(stmt)

return render_template("profile.html",account=account)
```

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

```
def choice():
```

```
    if request.method == 'POST':
```

```
        choice1 = request.form['choice1']
```

```
        choice2 = request.form['choice2']
```

```
        choice3 = request.form['choice3']
```

```
    print(choice1,choice2,choice3)
```

```
    insert_sql = "INSERT INTO user VALUES (?, ?, ?, ?, ?, ?, ?, ?)"
```

```
    prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
    ibm_db.bind_param(prepare_stmt, 1, session['email'])
```

```
    ibm_db.bind_param(prepare_stmt, 2, session['name'])
```

```
    ibm_db.bind_param(prepare_stmt, 3, session['phone'])
```

```
    ibm_db.bind_param(prepare_stmt, 4, session['country'])
```

```
    ibm_db.bind_param(prepare_stmt, 5, session['password'])
```

```
    ibm_db.bind_param(prepare_stmt, 6, choice1)
```

```
    ibm_db.bind_param(prepare_stmt, 7, choice2)
```

```
    ibm_db.bind_param(prepare_stmt, 8, choice3)
```

```

        ibm_db.execute(prepare_stmt)

    session.pop('phone', None)
    session.pop('country', None)
    session.pop('password', None)

    return redirect('/home')

    return render_template("choice.html")
@app.route('/logout')
def logout():
    session.pop('email', None)

    return redirect('/')
if __name__=="__main__":
    app.run(host='0.0.0.0',debug=True)
MAIL.PY
# using SendGrid's Python Library
# https://github.com/sendgrid/sendgrid-python
import os

from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
from random import randint

def send_mail(usermail,usersubject,usercontent,otp):
    if otp == "yes":

```

```
        content = '<strong>YOUR OTP:</strong>'+str(usercontent)
else:
    content = usercontent

message = Mail(
    from_email='mvishal2121@gmail.com',
    to_emails=str(usermail),
    subject=str(usersubject),
    html_content=str(content))
try:
    sg = SendGridAPIClient("")
    response = sg.send(message)
    print(response.status_code)
    print(response.body)
    print(response.headers)
except Exception as e:
    print(e.message)

#send_mail("vishalmadeshware27@gmail.com","Otp from news
tracker","1234","yes")
```

```
def generate_number():
    number=randint(1001,9999)
    return number
```

```
#print(generate_number())
```

## **HEADLINES.PY**

```
from newsapi import NewsApiClient
```

```
import pycountry

def news_category(country,category):

    # you have to get your api key from newapi.com and then paste it below
    newsapi = NewsApiClient(api_key='30b9110c469b415d97ecc0445cc74ba3')

    # now we will take name of country from user as input
    input_country = country
    input_countries = [f'{input_country.strip()}']
    countries = { }

    # iterate over all the countries in
    # the world using pycountry module
    for country in pycountry.countries:

        # and store the unique code of each country
        # in the dictionary along with it's full name
        countries[country.name] = country.alpha_2

    # now we will check that the entered country name is
    # valid or invalid using the unique code
    codes = [countries.get(country.title(), 'Unknown code')
              for country in input_countries]

    # now we have to display all the categories from which user will
    # decide and enter the name of that category
```

```
option = category

# now we will fetch the new according to the choice of the user
top_headlines = newsapi.get_top_headlines(

    # getting top headlines from all the news channels
    category=f'{option.lower()}', language='en',
country=f'{codes[0].lower()}')

# fetch the top news under that category
Headlines = top_headlines['articles']
title=[]
url=[]
urltoimage=[]

# now we will display the that news with a good readability for user
if Headlines:

    for articles in Headlines:

        b = articles['title'][:-1].index("-")
        if "news" in (articles['title'][-b+1:]).lower():
            title.append(f'{articles["title"]}')
            url.append(f'{articles["url"]}')

urltoimage.append(f'{articles["urlToImage"]}')
```



```

        else:

            title.append(f"{articles['title']}")

            url.append(f"{articles['url']}")

urltoimage.append(f"{articles['urlToImage']}")

        return title,url,urltoimage

    else:

        print(f"Sorry no articles found for {input_country},
Something Wrong!!!")

        option = input("Do you want to search again[Yes/No]?")

def news_headlines(country):

    # you have to get your api key from newapi.com and then
paste it below

    newsapi = NewsApiClient(api_key="")

    # now we will take name of country from user as input
    input_country = country
    input_countries = [f'{input_country.strip()}']
    countries = {}

    # iterate over all the countries in
    # the world using pycountry module
    for country in pycountry.countries:

        # and store the unique code of each country

```

```
# in the dictionary along with it's full name
countries[country.name] = country.alpha_2
```

```
# now we will check that the entered country name is
# valid or invalid using the unique code
codes = [countries.get(country.title(), 'Unknown code')
          for country in input_countries]
```

will

```
# now we have to display all the categories from which user
```

```
# decide and enter the name of that category
```

user

```
# now we will fetch the new according to the choice of the
```

```
top_headlines = newsapi.get_top_headlines(
```

```
# getting top headlines from all the news channels
language='en', country=f'{codes[0].lower()}')
```

```
# fetch the top news under that category
```

```
Headlines = top_headlines['articles']
```

```
title = []
```

```
url = []
```

```
urltoimage = []
```

for user

```
# now we will display the that news with a good readability
```

```

        if Headlines:
            for articles in Headlines:
                b = articles['title'][:-1].index("-")
                if "news" in (articles['title'][-b + 1:]).lower():
                    title.append(f"{articles['title']}")
                    url.append(f"{articles['url']}")

                urltoimage.append(f"{articles['urlToImage']}")

            else:
                title.append(f"{articles['title']}")
                url.append(f"{articles['url']}")

                urltoimage.append(f"{articles['urlToImage']}")

            return title, url, urltoimage

        else:
            print(f"Sorry no articles found for {input_country},
Something Wrong!!!")

            option = input("Do you want to search
again[Yes/No]?")

```

## **KEYSEARCH.PY**

```

import requests

def search_by_key(keyword):
    url = "https://newscatcher.p.rapidapi.com/v1/search"

```

```
querystring =  
{ "q":keyword,"lang":"en","sort_by":"relevancy","page":"1","media":"True" }
```

```
headers = {  
  
    "X-RapidAPI-Key": "",  
    "X-RapidAPI-Host": "newscatcher.p.rapidapi.com"  
  
}
```

```
response = requests.request("GET", url, headers=headers,  
params=querystring).json()
```

```
title=[]  
url=[]  
url_img=[]  
for i in response['articles']:  
    title.append(i['title'])  
    url.append(i['link'])  
    url_img.append(i['media_content'][0])
```

```
return title,url,url_img
```

## **PROJECT DEMO LINK**

[https://ibm-project-pnt2022tmid01417.s3.jp-tok.cloud-object-storage.appdomain.cloud/video3925571355\\_AdobeExpress%20\(1\)%20\(1\)%20\(1\).mp4](https://ibm-project-pnt2022tmid01417.s3.jp-tok.cloud-object-storage.appdomain.cloud/video3925571355_AdobeExpress%20(1)%20(1)%20(1).mp4)

[https://drive.google.com/file/d/1Bu5Bkfeapnsr-ktBxEF9PVhT68-Hu42y/view?usp=share\\_link](https://drive.google.com/file/d/1Bu5Bkfeapnsr-ktBxEF9PVhT68-Hu42y/view?usp=share_link)

## **GITHUB LINK**

<https://github.com/IBM-EPBL/IBM-Project-5108-1658747825>

