

NEWS TRACKER APPLICATION USING IBM CLOUD

A PROJECT REPORT

Submitted by

ANDREW G SELWYN

THIRUGNANASAMBANTHAM K

KINGSLY KIRUBAKARAN L

BEVIN NANTHA KUMAR V

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

ELCETRONICS AND COMMUNICATION ENGINEERING

UNIVERSITY VOC COLLEGE OF ENGINEERING

THOOTHUKUKUDI CAMPUS

ANNA UNIVERSITY:: CHENNAI 600025

NOV 2022

ANNA UNIVERSITY :CHENNAI 600025

BONAFIDE CERTIFICATE

Certificate that this project report “NEWS TRACKER APPILCATION” is the bonafide work of “ANDREW G SELWYN (953219106004), THIRUGNANASAMBANTHAM K (953219106040), KINGSLYKIRUBAKARAN L (953219106014), BEVIN NANTHA KUMAR V (953219106007)” who carried out the project work under my supervision.

SIGNATURE

Dr.A.MOOKAMBIGA

MENTOR

Asst.PROFESSOR, ECE

University VOC College of Engineering,

Thoothukudi.

CONTENTS

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:

The application is exclusively for the news reader's who need the whole update in their hand. It consists of various news on the daily basis which can be easily readable for the user's. The whole regional news can be covered with each domain. There is many popular people's account in our application and the people can be followed to get their daily updates.

Integrate the application with news APIs and store the data in the database. The database will have all the details and the user can search the news by using a search bar.

1.2 Purpose:

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

2. LITERATURE SURVEY:

2.1 Existing problem:

The existing system has many Features which is only implemented in some countries. The system is made up of HTML, CSS and Javascript. The existing system provide many services to the user based on their needs. The system contains every news which user can sort by their own. The system has an account of popularity person in order to follow them and to know their news.

2.2 References :

PAPER 1

TITLE: Exploring mobile news reading interactions for news app Personalization

AUTHOR: Marios Constantinides, John Dowell.

DESCRIPTION:

As news is increasingly accessed on smartphones and tablets, the need for personalising news app interactions is apparent.

PAPER 2:

TITLE: Survey on Virtual Assistant

AUTHOR: Amrita Sunil Tulshan

DESCRIPTION:

Many services were covered by these assistants but still there are some improvements required in voice recognition, contextual understanding and hand free interaction

2.3 Problem Statement Definition:

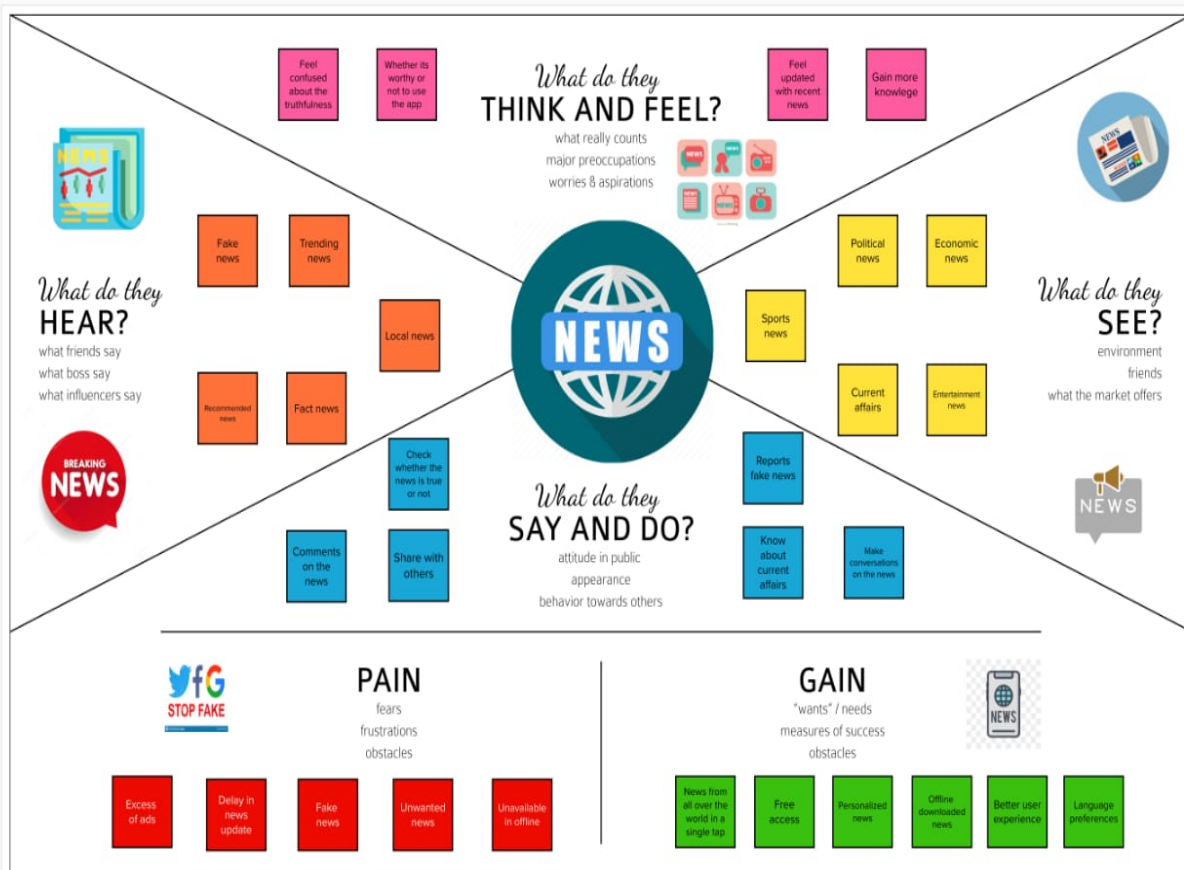
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	I am Student	I'm trying to get quick access of technology news	But the application push the irrelevant news	Because of poor optimization	Which makes me feel annoying
PS-2 I	I am blogger	I'm trying to read specific News articles	But it displays unrelated articles	Because of poor optimized filter option	Which makes me feel disappointed

Customer Problem Statement:

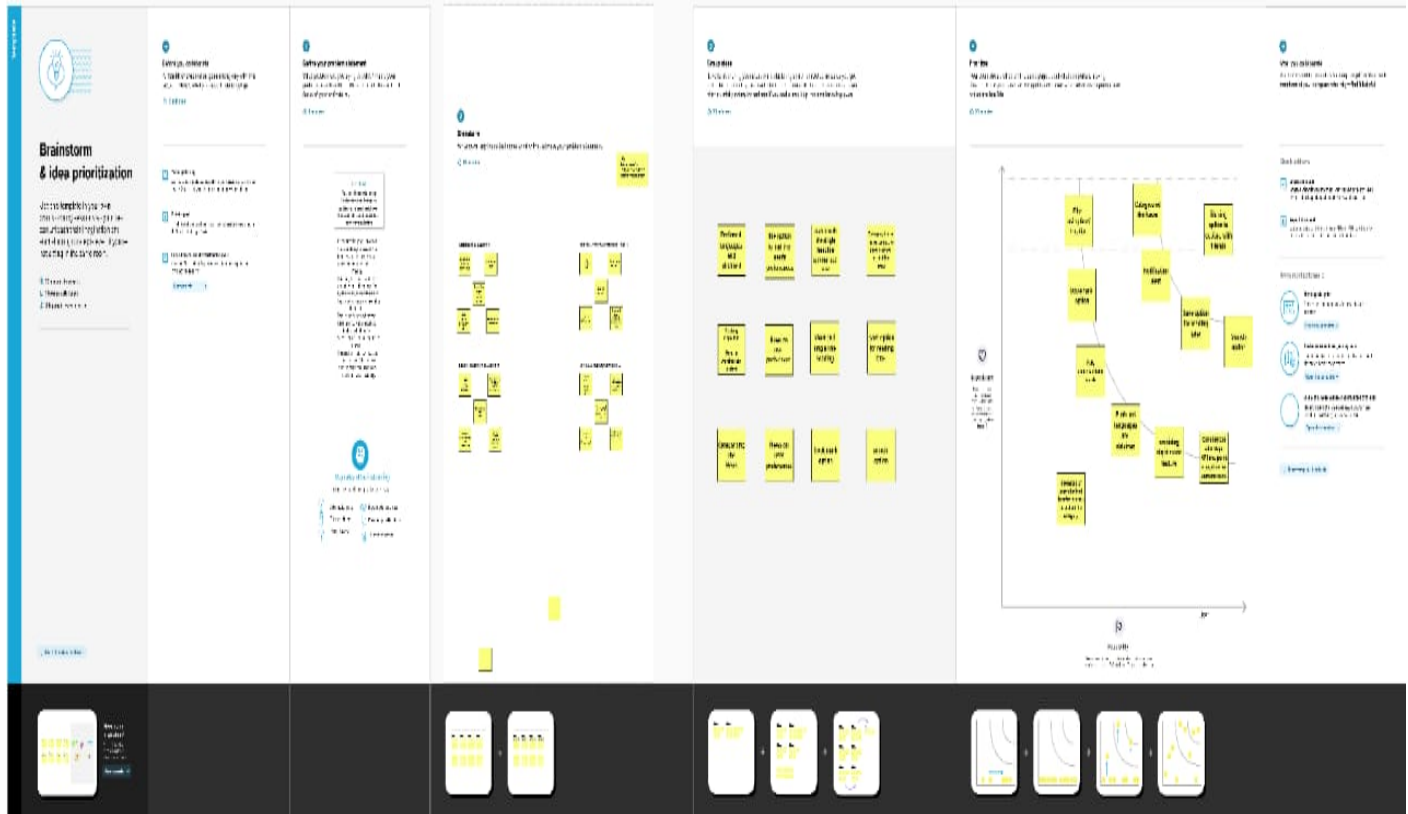
A well-articulated customer problem statement allows us to find the ideal solution for the challenges our customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive the news. There may be many fake news and the news can be filtered by news api.

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:



3.3 Proposed Solution:

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	To keep track of daily news using Cloud Computing.
2	Idea / Solution description	<p>☒ Nowadays most of the people do not have the time to read a newspaper in their daily life.</p> <p>☒ But they have to keep updated about the incidents occurring in their surroundings.</p> <p>☒ So our app is very much useful to know daily news in mobile phones or PC.</p>
3	Novelty / Uniqueness	<p>☒ It is not like any other apps, it shows relevant information and shows only what we need.</p> <p>☒ With the help of our customer location, the app can automatically post the news in their surroundings.</p>
4	Social Impact / Customer Satisfaction	By using our application, customers can read and even they can hear the news with the help of the voice assistant.
5	Business Model (Revenue Model)	Money for buying newspaper can be saved and we can update the same information in our mobile phones or PC.
6	Scalability of the Solution	<p>☒ Whatever the news we want, we can get the information immediately within a moment.</p> <p>☒ This makes our app more scalable.</p>

3.4 Problem Solution fit:

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) Who is your customer? i.e. working parents of 0-5 y.o. Kids CS The report of or on a news event occurring within 24 hours prior to the broadcast or other distribution thereof by means of video, film, photography or other sound and image reproduction technology that has been produced and edited into a news story.	6. CUSTOMER CONSTRAINTS 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. CC The main constraints are memory, battery life, ability to adapt to different screen sizes and orientations, security, and network bandwidth.	5. AVAILABLE SOLUTIONS 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 note taking AS Harnessing wide API to incorporate regular updating of news. The user interacts with the application. Integration the application with news APIs and store the data in the database.	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. J&P By collecting relevant information from the user through app. Cloud is used to store and transmit the data by using DB2. Weather APIs are employed to assist the user in making decision. The user could take decision through a mobile application.	9. PROBLEM ROOT CAUSE 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. RC As our lives are busy nowadays, we often feel we need more than 24 hours. A day to cope up with every this to be updated our self via news in our schedule. Well, that's not possible but reducing the time by using the conventional method of reading news can help at any time in any place to be updated.	7. BEHAVIOUR 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) BE User Registration & login via email & social profiles. Browse News Categories. Search Bar is given. Frequently searched message sent through email and SMS. Underline & save text. Save news and quotes for reading later. Machine learning based news on user preferences. Social integration for sharing news. Push notifications and news alerts via emails and SMS. Fake news will be detected.	
Focus on J&P, tap into BE, understand RC	3. TRIGGERS What triggers customers to act? i.e. seeing their neighbor installing solar panels, reading about a more efficient solution in the news TR User facing issues in providing proper information about the news. No proper internet connection that leads to reduced speed of live news which affects the user's time.	10. YOUR SOLUTION 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 behavior. SL Our app collects the relevant information and sends news to the main server. It also collects the weather data from the weather API. Feasible UI to deploy the application in simple solution	8. CHANNELS of BEHAVIOR 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 CH Apps allow the user to perform a task better with the help of the gesture. It is very easy to navigate through, unlike the website. A well designed application that can perform fast. 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. Users can access records, collect data and store images or access manuals without cell service or a Wi-Fi connection.	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER 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. EM BEFORE: Lack of knowledge in cloud computing Random decisions AFTER: Data from reliable sources Correct decision			

4. REQUIREMENT ANALYSIS:

4.1 Functional requirement:

Following are the functional requirements of the proposed solution.

S.No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	User Installation	User can install the application through playstore
2	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
3	User Confirmation	Confirmation via Email Confirmation via OTP
4	User Login	Login using ID and password
5	User Interaction	User will receive the news according to his interests.
6	User Logout	User can completely log out from device and it is not mandatory.

4.2 Non-Functional requirements:

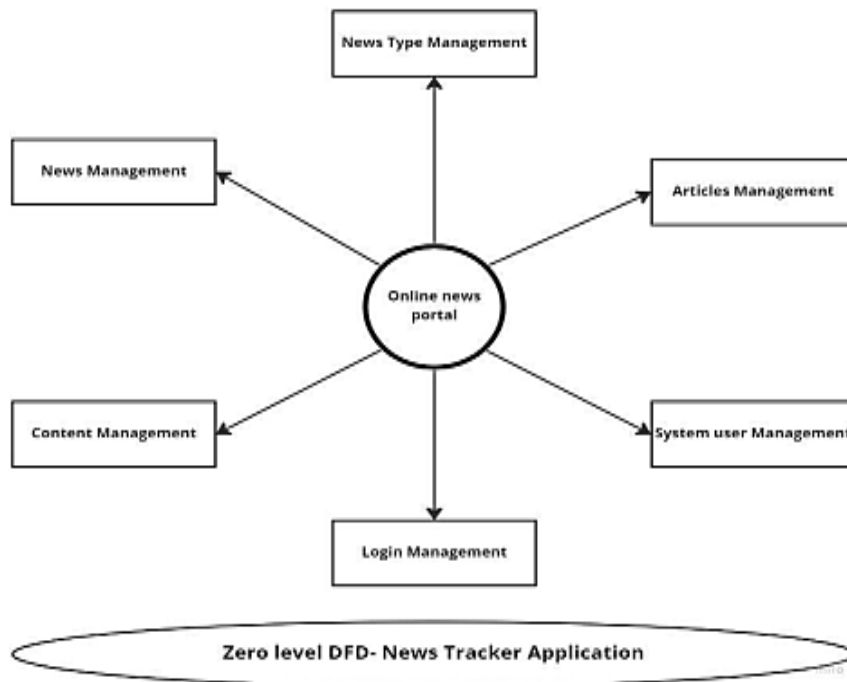
Following are the non-functional requirements of the proposed solution.

S.No	Non-Functional Requirement	Description
1	Usability	The usability of the application is very high because the number of users are very high and everyone wants to know the accurate news.
2	Security	The information in user's profile is completely safe with strong passwords. The published news can't be modified to spread fake news.
3	Reliability	The application should deliver only reliable content from trusted sources.
4	Performance	The performance of the app should be perfect. It should be smooth and interactive without any issues.
5	Availability	Continuous availability of our service must be provided all the time with the support of internet.
6	Scalability	The application should run easily with 5000 users at the same time. The Scalability will be increased by increase of users.

5. PROJECT DESIGN :

5.1 Data flow diagram:

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:

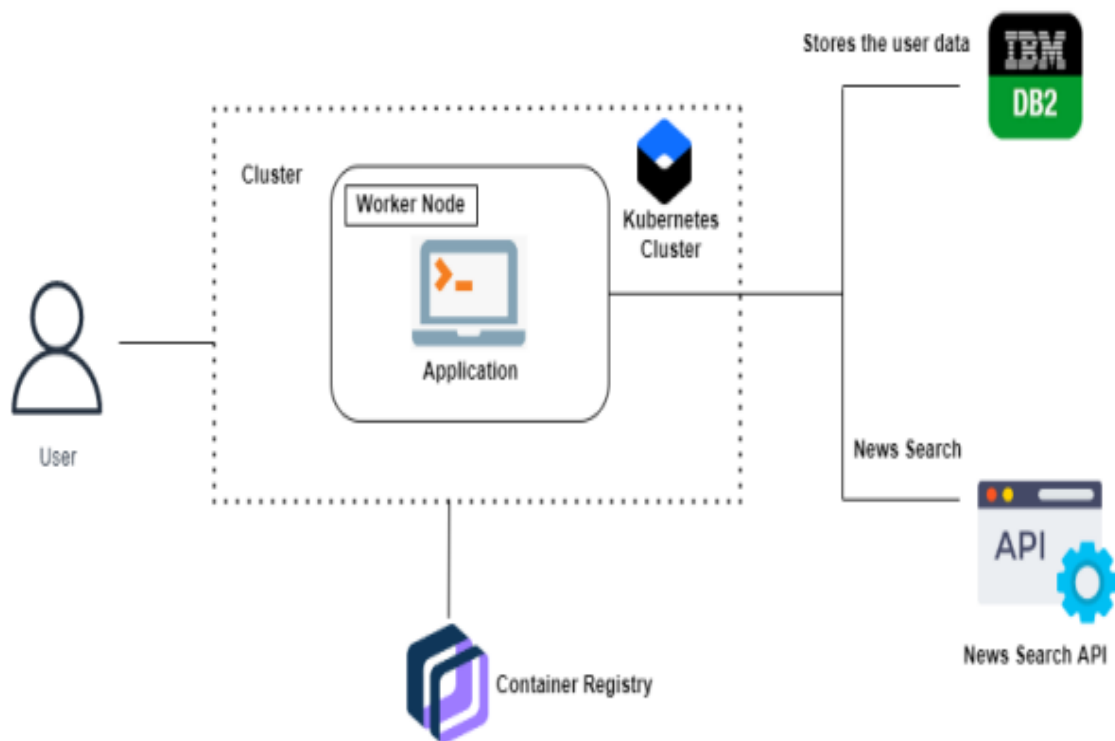


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	The user can interact with the application to know about the trending news	HTML, CSS, JavaScript / React Js etc.
2	Application Logic-1	The application contains this resource gives you basic understanding of Flask.	Python Flask.
3	Application Logic-2	The application contains the news sub-division like business news, economic news and general news	IBM Watson STT service
4	Database	User details are getting stored in the MySQL database.	MySQL, etc.
5	Cloud Database	With the use of cloud , media coverage issue cannot be occurred and the images can be accessed via cloud	IBM DB2.

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 (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering email, password, and name	I can access my account dashboard	High	Sprint-1
		USN-2	As a user, I will redirected to the login page	I can access the web application	High	Sprint-1
		USN-3	As a user, I will register for the application through Gmail account	I can register & access the dashboard with Username login	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering Username & password	I can view all types of information through this application	High	Sprint-1
	Dashboard	USN-1	As a user, I can log into the application and look into my dashboard	I can Look into My Dashboard after my login	Low	Sprint-1
		USN-2	As a user, I can log into the application and update my personal data	I can view the personal data which can be updated by the user	Low	Sprint-2
		USN-3	As a user, I can log into the application and read news based on my filter contents	News contents are filtered based on the user needs	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING:

Project Tracker, Velocity & Burn down Chart:

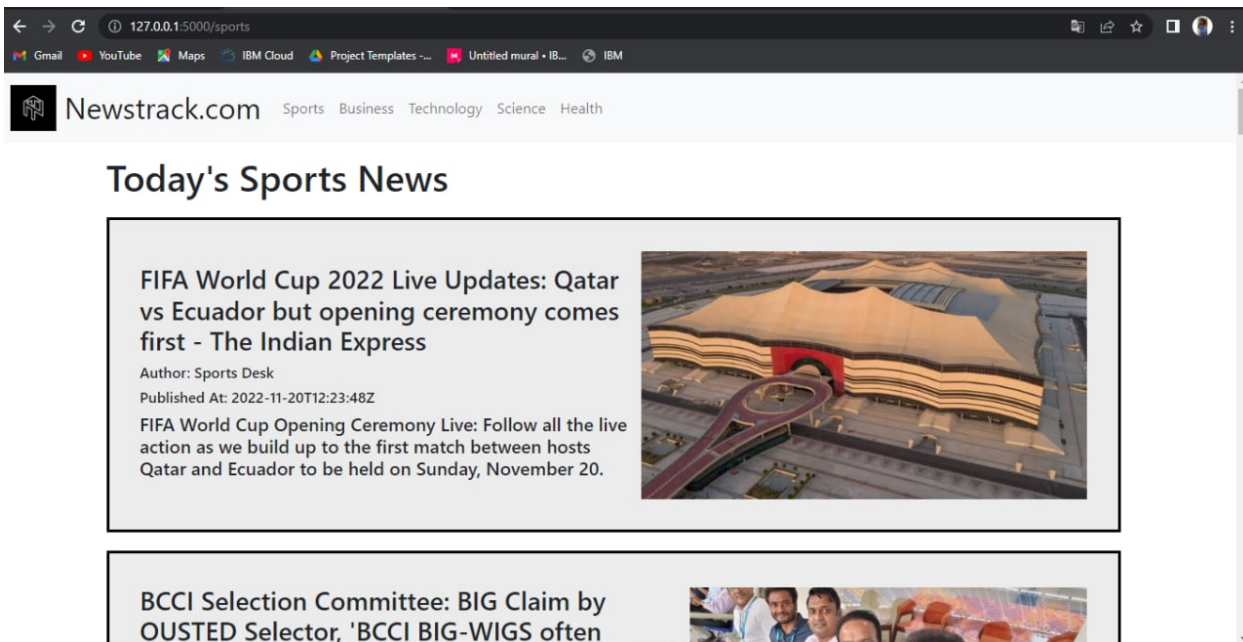
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	01 Nov 2022	07 Nov 2022	20	05 Nov 2022
Sprint-2	20	5 Days	06 Nov 2022	07 Nov 2022	20	10 Nov 2022
Sprint-3	20	5 Days	02 Nov 2022	08 Nov 2022	20	15 Nov 2022
Sprint-4	20	4 Days	02 Nov 2022	08 Nov 2022	20	19 Nov 2022

Sprint Delivery Schedule:

Velocity We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint).
Calculating the team's average velocity (**AV**) per iteration unit (story points per day)

$$AV = \text{sprint duration/velocity} = 20/6 = 3.33$$

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



Today's Business News

First Hero-Harley co-developed bike to debut in 2024 - Team-BHP

Author: Tushar Kelshikar

Published At: 2022-11-20T08:40:54Z

Hero MotoCorp and America's Harley-Davidson have partnered to develop a range of premium motorcycles. It is said that their first co-developed bike could debut by the end of March 2024. Niranjn Gupta, CFO of Hero MotoCorp, has stated that the company would l...



Harsh Goenka Says People Are Underestimating Elon Musk's "Genius" Calls



Today's Tech News

YouTube Streamer Dr Disrespect Gives Transparent Views for This GOTY Nominee, Is It God of War: Ragnarok or... - EssentiallySports

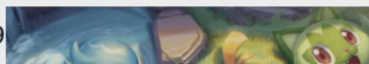
Author: None

Published At: 2022-11-20T11:50:00Z

YouTube streamer Dr Disrespect has stated his views about the potential Game of the Year. Is it God of War: Ragnarok or another game?



Pokemon Scarlet and Violet: What TMs can Gen 9 starters Sprigatito, Quaxly, and Fuecoco learn?





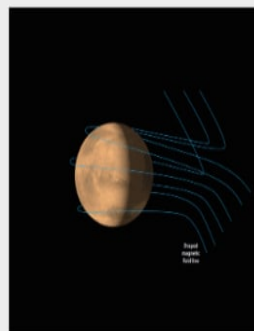
Today's Science News

Double auroras light up Martian skies - EarthSky

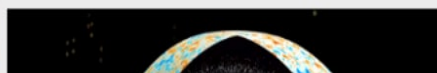
Author: None

Published At: 2022-11-20T11:36:24Z

None



New Map of the Universe Displays Span of Entire Cosmos With Pinpoint Accuracy and



Today's health News

Health and Wellness: Antibiotic resistance - Des Moines Register

Author: Ann Cochran - Dallas County Public Health, The Des Moines Register

Published At: 2022-11-20T12:33:18Z

Each time you take antibiotics, you increase the chances the bacteria in your body will develop a resistance to that medication. Resistant infections may become less and less treatable, and therefore...

Image
Not
Found

Collagen for anti-aging: Do you know these facts about collagen supplements? - Health shots

Author: Purvi Kalra

Published At: 2022-11-20T12:30:41Z



9. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- ☒ Easily track the news
- ☒ Reduce paper works
- ☒ Limitations in spending
- ☒ Manage every news dynamically
- ☒ Well suitable for urging people

DISADVANTAGES:

- ☒ More dependent on mobile phones
- ☒ Initially difficult to understand for uneducated people

10. CONCLUSION:

The purpose of the project is to develop an application which will display news articles and videos verified and approved at backend. User can follow the eminent personality accounts to get their related news daily.

The scope of the project includes followings:

Functional specification document to describe the complete system in low level. User interface design for website and mobile phone for both iOS and Android. Development of all functional requirements as proposed for the mobile application. Setting up the system on live server in ready for use condition.

11. FUTURE SCOPE:

The Existing News tracker website or application have basic user interface which lag on some areas and the news were dynamically changed according to the time.

In future, we are planning to add advertisements which make high revenue.

☒ User can browse news articles and videos on various categories by searching through searchbox or voice assistance.

☒ The voice recognition can be used as an easier source even for uneducated people to track the news so that they can able see what they need.

12. APPENDIX:

Source Code:

Index.html

```
{% extends 'base.html' %}
{% block body %}
<div class="container">
  <h1>Today's News</h1>
  {% for news in cases.articles %}
    <div class="news_box">
      <div class="info_div">
        <h3 class="title">{{news.title}}</h3>
        <h6>Author: {{news.author}}</h6>
        <h6>Published At: {{news.publishedAt}}</h6>
        <h5>{{news.description}}</h5>
      </div>
      <div class="img_div">
        
      </div>
    </div>
  {% endfor %}
</div>
{% endblock body %}
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="static/css/style.css">
  <link href='https://unpkg.com/boxicons@2.1.2/css/boxicons.
min.css' rel='stylesheet'>
  <title>Login</title>
```

```
</head>
```

```
<style>
```

```
/* Google Fonts - Poppins */
```

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap');
```

```
*{  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
  font-family: 'Poppins', sans-serif;  
}
```

```
.container{  
  height: 85vh;  
  width: 100%;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  column-gap: 30px;  
}
```

```
.top{  
  width: 100%;  
  height: 90px;  
  background: linear-gradient(to right, rgb(50, 32, 51) 0%, rgb(118, 50, 69) 90%);  
  display: flex;  
  align-items: center;  
  position: fixed;  
}
```

```
.top h3{  
  color: rgb(178, 183, 230);  
  font-size: 25px;  
  display: block;  
  width: 500px;  
  margin-left: 10px;  
}
```

```
.right{  
  width: 100%;  
  height: 100%;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
.right form{  
  transition: 2s;  
  width: 250px;  
  height: 270px;  
  background: rgb(27, 61, 75);  
  padding: 48px;  
  border-radius: 10px;  
  margin-top: 120px;  
}
```

```
.right form h2{  
  text-align: center;  
  color: rgb(255, 255, 255);  
}
```

```
.right form input{  
  font-family: Arial, Helvetica, sans-serif;  
  color: rgb(245, 239, 239);  
  height: 25px;  
  margin-top: 20px;  
  background: none;  
  border: none;  
  border-bottom: 1px solid rgba(218, 165, 32, 0.445);  
  padding-left: 10px;  
  width: 160px;  
}
```

```
#btn{  
  width: 90px;  
  height: 40px;  
  background-color: rgb(19, 105, 139);  
  border-radius: 5px;
```

```
position: relative;
left: 30px;
border: none;
top: 10px;
}
}
```

```
.right form:hover #btn{
background:white;
color: rgb(19, 125, 139);
cursor: pointer;
}
#btn:hover{
background-color: black;
}
.right form:hover input{
border-bottom: 1px solid rgba(5, 4, 2, 0.445);
```

```
}
.right form:hover input::placeholder{
color: black;
}
.reg{
width: 100%;
height: 90px;
display: flex;
align-items: center;
}
```

```
.right a{
text-decoration: none;
color: black;
background-color: rgba(0, 195, 255, 0.616);
display: block;
width: 120px;
height: 40px;
border-radius:5px;
```

```
    text-align: center;
    padding-top: 10px;
    position: absolute;
    right: 30px;
    top: 30px;
}
input::placeholder{
    color: rgba(255, 255, 255, 0.521);
    font-weight: lighter;
    font-size: 13px;
}
.form{
    position: absolute;
    max-width: 430px;
    width: 100%;
    padding: 30px;
    border-radius: 6px;
    background: linear-gradient(to top, rgba(29, 18, 35, 255) 0%, rgba(55, 34, 51, 255),
    rgba(248, 160, 158, 255), rgba(212, 86, 98, 255) , rgba(73, 39, 63, 255) 90%);
}

.header{
    font-size: 28px;
    font-weight: 600;
    color: rgba(222, 82, 83, 255);
    text-align: center;
}
.form{
    margin-top: 30px;
}
.form .field{
    position: relative;
    height: 50px;
    width: 100%;
    margin-top: 20px;
    border-radius: 6px;
}
.field input,
.field button{
```



```
height: 100%;  
width: 100%;  
border: none;  
font-size: 16px;  
font-weight: 400;  
border-radius: 6px;  
}  
.field input{  
outline: none;  
padding: 0 15px;  
border: 1px solid#CACACA;  
}  
.field input:focus{  
border-bottom-width: 2px;  
}  
.eye-icon{  
position: absolute;  
top: 50%;  
right: 10px;  
transform: translateY(-50%);  
font-size: 18px;  
color: #8b8b8b;  
cursor: pointer;  
padding: 5px;  
}  
.field button{  
color: #fff;  
background-color: rgba(247,142,139,255);  
transition: all 0.3s ease;  
cursor: pointer;  
}  
.field button:hover{  
background-color:rgba(60,36,57,255);  
}  
.form-link{  
text-align: center;  
margin-top: 10px;  
}  
.form-link span,
```

```
.form-link a{
  font-size: 14px;
  font-weight: 400;
  color: white;
}

.form a{
  color: yellow;
  text-decoration: none;
}

.form-content a:hover{
  text-decoration: underline;
}

.line{
  position: relative;
  height: 1px;
  width: 100%;
  margin: 36px 0;
  background-color: #d4d4d4;
}

.line::before{
  content: 'Or';
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background-color: #FFF;
  color: #8b8b8b;
  padding: 0 15px;
}

.media-options a{
  display: flex;
  align-items: center;
  justify-content: center;
}

a.facebook{
  color: #fff;
  background-color: #4267b2;
}

a.facebook .facebook-icon{
```

```
height: 28px;
width: 28px;
color: #0171d3;
font-size: 20px;
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
background-color: #fff;
}
.facebook-icon,
img.google-img{
position: absolute;
top: 50%;
left: 15px;
transform: translateY(-50%);
}
img.google-img{
height: 20px;
width: 20px;
object-fit: cover;
}
a.google{
border: 1px solid white;
}
a.google span{
font-weight: 500;
opacity: 0.6;
color: white;
}

@media screen and (max-width: 400px) {
.form{
padding: 20px 10px;
}
}
</style>
```

```
<body style = "background-image: url('static/images/red-mountains-
morning-minimal.jpg');
background-attachment: fixed; background-size: 100% 100%;">

  <div class="top">
    <h3>Newstrack.com</h3>
  </div>

  <section class="container forms">
    <div class="form login">
      <div class="form-content">
        <div class="right">
          <div class="form">
            <form action="/login" method="post">
              <div class="msg">{{ msg }}</div>
              <h2>LOGIN</h2>
              <input type="text" name="username" placeholder="Username"
                id="username"><br>
              <input type="password" name="password" placeholder="Password"
                id="password"><br>
              <input type="submit" id="btn" value="LOGIN">
            </form>
          </div>
          <div class="form-link">
            <span>Don't have an account? <a href="register">
              Register</a></span>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div class="line"></div>

  <div class="media-options">
    <a href="#" class="field facebook">
      <i class='bx bxl-facebook facebook-icon'></i>
      <span>Login with Facebook</span>
    </a>
  </div>

  <div class="media-options">
    <a href="#" class="field google">
      
        <span>Login with Google</span>
    </a>
</div>

</div>
</section>
</body>
</html>

<script>
    window.watsonAssistantChatOptions = {
        integrationID: "cd6753a2-76fe-4af5-b73c-c7d998962504", //
        The ID of this integration.
        region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "2e663844-6d94-4320-a6e0-79d3a8e2b775", //
        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>
<script src="static/js/script.js"></script>

```

Register.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="register.css">

```

```
<title>Register</title>
</head>
<style>
*{
margin:0;
padding:0;
font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial,
sans-serif;
box-sizing: border-box;
}

.container{
width: 100%;
height: 100vh;
background-color:aquamarine;
background-size: cover;
overflow-y: hidden;
display: flex;
}

.top{
width: 100%;
height: 90px;
background: linear-gradient(to right,rgb(0, 0, 0) 0%,rgba
(49, 42, 3, 0.945) 90%);
display: flex;
align-items: center;
position: fixed;
}

.top h3{
color: rgb(178, 183, 230);
font-size: 25px;
display: block;
width: 500px;
margin-left: 10px;
}

.right{
width: 100%;
```

```
height: 100%;
display: flex;
align-items:center;
justify-content: center;
}
.right form{
transition:2s;
width: 450px;
height: 470px;
background: linear-gradient(to bottom, rgba(32,41,134,255) 10%,
rgba(31,84,161,255), rgba(33,50,140,255), rgba(17,157,192,255),
rgba(117,206,247,255) 90%);
padding: 48px;
border-radius: 10px;
margin-top: 120px;
padding-left: 90px;
}
.right form h2{
text-align: center;
color: black;
font-size: 30px;
margin-right: 200px;
}
.right form input{
font-family: Arial, Helvetica, sans-serif;
color: rgb(233, 233, 233);
height: 25px;
margin-top: 30px;
background:none;
border: none;
border-bottom: 1px solid rgba(0, 0, 0, 0.445);
padding-left: 6px;
width: 280px;
}
.right form:hover h2{
color: rgb(168, 190, 201);;
```

```
text-align: center;
margin-right: 200px;
}
#btn{
width: 120px;
height: 40px;
background-color: rgb(31, 2, 65);
border-radius: 5px;
position: relative;
left: 70px;
border: none;
top: -10px;
color: white;
}
.right form:hover #btn{
background: linear-gradient(to right top, rgba(30, 7, 241, 0.788)
0%, rgba(30, 7, 241, 0.788) 60%);
color: rgb(14, 209, 235);
cursor: pointer;
}
#btn:hover{
background-color: black;
}
.right form:hover input{
border-bottom: 1px solid black;
}
.right form:hover input::placeholder{
color: rgb(247, 250, 252);
}
.reg{
width: 100%;
height: 100px;
display: flex;
align-items: center;
}
.right #go{
text-decoration: none;
```



```
    color: black;
    background-color: rgba(0, 162, 255, 0.616);
    display: block;
    width: 120px;
    height: 40px;
    border-radius: 5px;
    text-align: center;
    padding-top: 10px;
    position: absolute;
    right: 30px;
    top: 30px;
}
input::placeholder{
    color: black;
    font-weight: lighter;
    font-size: 13px;
}
```

```
</style>
```

```
<body>
```

```
    <div class="top">
```

```
        <h3>Newstrack.com</h3>
```

```
    </div>
```

```
    <div class="container" style="background-image: url(static/images
    /reg2.jpg);background-attachment: fixed;background-size:100% 100%;">
```

```
        <div class="right">
```

```
            <a href="backlogin" id="go">Go Back</a>
```

```
            <form action="/register" method="post">
```

```
                <div class="msg">{{ msg }}</div>
```

```
                <h2>REGISTER</h2>
```

```
                <input type="text" name="username" placeholder="Usernam
                e"><br>
```

```
                <input type="mail" name="email" placeholder="Mail id"
                ><br>
```

```
                <input type="number" name="phone_num" placeholder="
                Phone Number"><br>
```

```

        <input type="password" name="password" placeholder=
        "Password"><br>
        <input type="password" name="confirm_password"
        placeholder="Confirm password"><br>
        <input type="submit" id="btn" value="REGISTER">
    
```

Science.html

```

{% extends 'base.html' %}
{% block body %}
<div class="container">
    <h1>Today's Science News</h1>
    {% for news in cases.articles %}
    <div class="news_box">
        <div class="info_div">
            <h3 class="title">{{news.title}}</h3>
            <h6>Author: {{news.author}}</h6>
            <h6>Published At: {{news.publishedAt}}</h6>
            <h5>{{news.description}}</h5>
        </div>
        <div class="img_div">
            
        </div>
    </div>
    {% endfor %}
</div>
{% endblock body %}

```

Sports.html

```

{% extends 'base.html' %}
{% block body %}

```

```

<div class="container">
  <h1>Today's Sports News</h1>
  {% for news in cases.articles %}
    <div class="news_box">
      <div class="info_div">
        <h3 class="title">{{news.title}}</h3>
        <h6>Author: {{news.author}}</h6>
        <h6>Published At: {{news.publishedAt}}</h6>
        <h5>{{news.description}}</h5>
      </div>
      <div class="img_div">
        
      </div>
    </div>
  {% endfor %}
</div>
{% endblock body %}

```

Tech.html

```

{% extends 'base.html' %}
{% block body %}
<div class="container">
  <h1>Today's Tech News</h1>
  {% for news in cases.articles %}
    <div class="news_box">
      <div class="info_div">
        <h3 class="title">{{news.title}}</h3>
        <h6>Author: {{news.author}}</h6>
        <h6>Published At: {{news.publishedAt}}</h6>
        <h5>{{news.description}}</h5>
      </div>
      <div class="img_div">
        
      </div>
    </div>
  {% endfor %}
</div>
{% endblock body %}

```

Health.html

```
{% extends 'base.html' %}
{% block body %}
<div class="container">
  <h1>Today's health News</h1>
  {% for news in cases.articles %}
    <div class="news_box">
      <div class="info_div">
        <h3 class="title">{{news.title}}</h3>
        <h6>Author: {{news.author}}</h6>
        <h6>Published At: {{news.publishedAt}}</h6>
        <h5>{{news.description}}</h5>
      </div>
      <div class="img_div">
        
      </div>
    </div>
  {% endfor %}
</div>
{% endblock body %}
```

Business.html

```
{% extends 'base.html' %}
{% block body %}
<div class="container">
  <h1>Today's Business News</h1>
  {% for news in cases.articles %}
    <div class="news_box">
      <div class="info_div">
        <h3 class="title">{{news.title}}</h3>
        <h6>Author: {{news.author}}</h6>
        <h6>Published At: {{news.publishedAt}}</h6>
        <h5>{{news.description}}</h5>
      </div>
      <div class="img_div">
```

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

Base.html

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

<head>

    <meta charset="utf-8">

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

    <link href="https://cdn.jsdelivr.net/npm/bootstrap
@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8
tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin=
"anonymous">

    <title>www.newstrack.com</title>

    <style>

        .container {

            margin-top: 20px;

        }

        h1{

            margin-top: 8%;

        }

        .nav{

            overflow: hidden;

            position: fixed;

            top: 0;

            width: 100%;

        }

        .news_box {

            border: 3px solid rgb(0, 0, 0);
```

```

margin-top: 20px;
margin-bottom: 20px;
margin-right: 10px;
padding: 3%;
display: flex;
align-items: center;
justify-content: space-between;
font-size: 1.2em;
background-color: rgb(236, 236, 236);
}

.navbar-brand {
text-decoration: none;
font-size: 2em;
}

@media screen and (max-width: 980px) {

.news_box {
flex-direction: column;
}

.img_div img{
width: 100%;
}
}

</style>
</head>
<body >
<nav class="navbar nav navbar-expand-lg navbar-light
bg-light">
<div class="container-fluid">
<a class="navbar-brand" href="{ { url_for('index') } }">

Newstrack.com</a>
<button class="navbar-toggler" type="button" data-
bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

```

```
<div class="collapse navbar-collapse" id="navbarNav">
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="{{ url_for('sports') }}">
        >Sports</a>
      </li>
    <li class="nav-item">
      <a class="nav-link" href="{{ url_for('business') }}">
        Business</a>
      </li>
    <li class="nav-item">
      <a class="nav-link" href="{{ url_for('technology') }}">
        Technology</a>
      </li>
    <li class="nav-item">
      <a class="nav-link" href="{{ url_for('science') }}">
        Science</a>
      </li>
    <li class="nav-item">
      <a class="nav-link" href="{{ url_for('health') }}">
        Health</a>
      </li>
    </ul>
  </div>
</div>
</nav>
{% block body %}

{% endblock body %}

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/
js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2Ml
QnikTlwXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="
anonymous"></script>
</body>
</html>
```

Static.css

Style.css

```
/* Google Fonts - Poppins */
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap');

*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}

.container{
  height: 100vh;
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  column-gap: 30px;
}

.form{
  position: absolute;
  max-width: 430px;
  width: 100%;
  padding: 30px;
  border-radius: 6px;
  background: linear-gradient(to top, rgba(29,18,35,255) 0%,
  rgba(55,34,51,255), rgba(248,160,158,255), rgba(212,86,98,255)
  , rgba(73,39,63,255) 90%);
}

.form.signup{
  opacity: 0;
  pointer-events: none;
}

.forms.show-signup .form.signup{
```



```
    opacity: 1;
    pointer-events: auto;
}
.forms.show-signup .form.login{
    opacity: 0;
    pointer-events: none;
}
header{
    font-size: 28px;
    font-weight: 600;
    color: rgba(222,82,83,255);
    text-align: center;
}
form{
    margin-top: 30px;
}
.form .field{
    position: relative;
    height: 50px;
    width: 100%;
    margin-top: 20px;
    border-radius: 6px;
}
.field input,
.field button{
    height: 100%;
    width: 100%;
    border: none;
    font-size: 16px;
    font-weight: 400;
    border-radius: 6px;
}
.field input{
    outline: none;
    padding: 0 15px;
    border: 1px solid#CACACA;
}
.field input:focus{
    border-bottom-width: 2px;
```

```
}  
.eye-icon{  
    position: absolute;  
    top: 50%;  
    right: 10px;  
    transform: translateY(-50%);  
    font-size: 18px;  
    color: #8b8b8b;  
    cursor: pointer;  
    padding: 5px;  
}  
.field button{  
    color: #fff;  
    background-color: rgba(247,142,139,255);  
    transition: all 0.3s ease;  
    cursor: pointer;  
}  
.field button:hover{  
    background-color: rgba(60,36,57,255);  
}  
.form-link{  
    text-align: center;  
    margin-top: 10px;  
}  
.form-link span,  
.form-link a{  
    font-size: 14px;  
    font-weight: 400;  
    color: white;  
}  
.form a{  
    color: yellow;  
    text-decoration: none;  
}  
.form-content a:hover{  
    text-decoration: underline;  
}  
.line{  
    position: relative;
```

```
height: 1px;
width: 100%;
margin: 36px 0;
background-color: #d4d4d4;
}
.line::before{
content: 'Or';
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
background-color: #FFF;
color: #8b8b8b;
padding: 0 15px;
}
.media-options a{
display: flex;
align-items: center;
justify-content: center;
}
a.facebook{
color: #fff;
background-color: #4267b2;
}
a.facebook .facebook-icon{
height: 28px;
width: 28px;
color: #0171d3;
font-size: 20px;
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
background-color: #fff;
}
.facebook-icon,
img.google-img{
position: absolute;
top: 50%;
```

```

    left: 15px;
    transform: translateY(-50%);
}
img.google-img{
    height: 20px;
    width: 20px;
    object-fit: cover;
}
a.google{
    border: 1px solid white;
}
a.google span{
    font-weight: 500;
    opacity: 0.6;
    color: white;
}

```

```

@media screen and (max-width: 400px) {
    .form{
        padding: 20px 10px;
    }
}

```

Js Script.js

```

const forms = document.querySelector(".forms"),
    pwShowHide = document.querySelectorAll(".eye-icon"),
    links = document.querySelectorAll(".link");

pwShowHide.forEach(eyeIcon => {
    eyeIcon.addEventListener("click", () => {
        let pwFields = eyeIcon.parentElement.parentElement
            .querySelectorAll(".password");

        pwFields.forEach(password => {
            if(password.type === "password"){
                password.type = "text";
            }
        })
    })
})

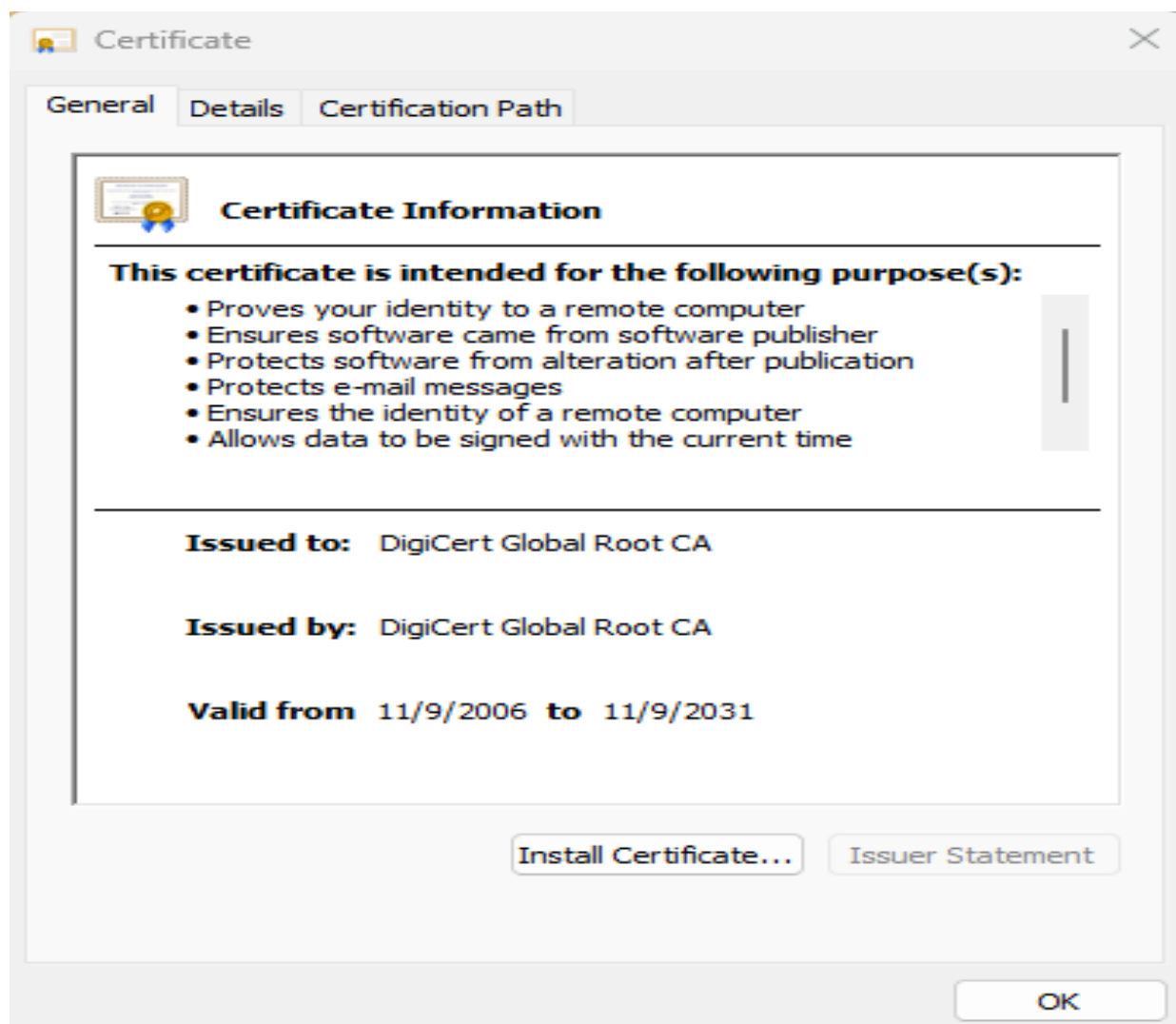
```

```

        eyeIcon.classList.replace("bx-hide", "bx-show");
        return;
    }
    password.type = "password";
    eyeIcon.classList.replace("bx-show", "bx-hide");
    })
}
    })
})

links.forEach(link => {
    link.addEventListener("click", e => {
        e.preventDefault(); //preventing form submit
        forms.classList.toggle("show-signup");
    })
})
})

```



App

App.py

```
from flask import*
from datetime import date
import ibm_db
import re
import requests

app=Flask(__name__)

app.secret_key = 'abcde'

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=fbd88901-ebdb
-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.app
domain.cloud;PORT=32731;SECURITY=SSL;SSLServerCertificate=
DigiCertGlobalRootCA.crt;UID=bgh46110;PWD=0FTZbkuJk6rYuFP7",'','')

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

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

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

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return render_template('login.html')
```

```

@app.route('/login', methods = ['GET', 'POST'])
def login():
    global userid
    msg = ''

    if request.method == 'POST' :
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM login WHERE username =? AND
        password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin'] = True
            session['id'] = account['USERNAME']
            userid= account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
        else:
            msg = 'Logged in successfully !'
            url = "https://newsapi.org/v2/top-headlines?country
            =in&category=general&apiKey=7e1737d3191d4fe894fc579df01b7bde"
            r= requests.get(url).json()
            case = {
                'articles': r['articles']
            }
            return render_template('index.html', cases = case)
        else:
            msg = 'Incorrect username / password !'
            return render_template('login.html', msg = msg)

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

```



```

def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        phone_num = request.form['phone_num']
        password = request.form['confirm_password']
        sql = "SELECT * FROM login WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
            return render_template('login.html', msg = msg)
        elif not re.match(r'^@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        elif not re.match(r'[0-9]+', phone_num):
            msg = 'phone number must contain only numbers !'
        else:
            insert_sql = "INSERT INTO user_details VALUES (?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, phone_num)
            ibm_db.bind_param(prepare_stmt, 4, password)
            ibm_db.execute(prepare_stmt)

            insert_sql_1 = "INSERT INTO login VALUES (?, ?)"
            prep_stmt_1 = ibm_db.prepare(conn, insert_sql_1)
            ibm_db.bind_param(prepare_stmt_1, 1, username)
            ibm_db.bind_param(prepare_stmt_1, 2, password)
            ibm_db.execute(prepare_stmt_1)
            msg = 'You have successfully registered !'
            return render_template('login.html', msg = msg)
    elif request.method == 'POST':

```

```
msg = 'Please fill out the form !'
```

```
return render_template('register.html', msg = msg)
```

```
return render_template('register.html', msg = msg)
```

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

```
def index():
```

```
url = "https://newsapi.org/v2/top-headlines?country=in&apiKey  
=660a62eb9a6445a7b887ad991e40f4cb"
```

```
r= requests.get(url).json()
```

```
case = {
```

```
    'articles': r['articles']
```

```
}
```

```
return render_template('index.html', cases = case)
```

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

```
def sports():
```

```
url = "https://newsapi.org/v2/top-headlines?country=in&category  
=sports&apiKey=660a62eb9a6445a7b887ad991e40f4cb"
```

```
r= requests.get(url).json()
```

```
case = {
```

```
    'articles': r['articles']
```

```
}
```

```
return render_template('sports.html', cases = case)
```

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

```
def business():
```

```
url = "https://newsapi.org/v2/top-headlines?country=in&category  
=business&apiKey=660a62eb9a6445a7b887ad991e40f4cb"
```

```
r= requests.get(url).json()
```

```
case = {
```

```
    'articles': r['articles']
```

```
}
```

```
return render_template('business.html', cases = case)
```

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

```
def technology():
```

```
url = "https://newsapi.org/v2/top-headlines?country=in&category
=technology&apiKey=660a62eb9a6445a7b887ad991e40f4cb"
r= requests.get(url).json()
case = {
    'articles': r['articles']
}
return render_template('tech.html',cases = case)

@app.route('/science')
def science():
    url = "https://newsapi.org/v2/top-headlines?country=in&category
=science&apiKey=660a62eb9a6445a7b887ad991e40f4cb"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('science.html',cases = case)

@app.route('/health')
def health():
    url = "https://newsapi.org/v2/top-headlines?country=in&category
=health&apiKey=660a62eb9a6445a7b887ad991e40f4cb"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('health.html',cases = case)

@app.route('/entertainment')
def entertainment():
    url = "https://newsapi.org/v2/top-headlines?country=in&category
=entertainment&apiKey=660a62eb9a6445a7b887ad991e40f4cb"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('entertainment.html',cases = case)

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

```
def search():
    s = request.form['search']
    today = date.today()
    dat = today.strftime("%Y/%m/%d")
    url = "https://newsapi.org/v2/everything?q="+s+"&from="+ dat+"
    &sortBy=popularity&apiKey=660a62eb9a6445a7b887ad991e40f4cb"
    r= requests.get(url).json()
    case = {
        'articles': r['articles']
    }
    return render_template('search.html', cases = case)

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

GitHub & Project Demo Link:

