

BRAINTHINK Report

Every page has a navigation where user can select options from;

- Blog
- About us
- Contact us

this navigation turns into a hamburger menu on tablet / mobile layout.

Header Colors are chosen to not disturb, and create a calm environment.

Body is set to a light blue to create contrast between articles and items on the page.

API End point url <https://excitable-breath.flywheelsites.com/>

The Homepage is Created using Fetch Url [https://excitable-breath.flywheelsites.com/wp-json/wp/v2/posts/\\${postId}](https://excitable-breath.flywheelsites.com/wp-json/wp/v2/posts/${postId})

Here it post all the postIds and adds it to the slider entered which are
`const postIds = [71, 69, 67, 65, 63, 61, 59, 57, 55, 53, 35, 33, 30];`

This makes it easy to either change posts to different Ids and also edit content which is located at Wordpress.

The slider is set to a autoloop every 25 seconds but will stop once an article is clicked and will start again once clicked outside of.

Under the slider is the main part of the page Here its created with normal HTML and fetch of right articles

Left articles and middle is written in

```
<article class="holder">
<a href="humanvsai.html"
    ></a>
<h2 class="title">
How good is enough? AI vs. human code
</h2>

<p>
    Published by Marc Luetkecke, Nikolas Rist and Tom Gehder
    on 16 Dec 2022/ Data-science, Training
  </p>
</article>
```

where the Ahref goes to a new HTML page where the specific ID is fetched and posted. Here the HTML structur is only written with a few lines.

```
<div id="results-container">
```

```
<div class="loading" id="loading"></div>
</div>
```

Left articles that are fetched is not displayed with images to not disturb and make it easier to read.

```
articleEl.innerHTML = `
```

```
<a href="${postId === 61 ? "amazon.html" : "development.html"}"><h1>${
title.rendered
}</h1></a>
```

```
<p>${excerptRendered}</p>
```

this code snippet generates HTML content for each article element, including a link, title, and excerpt, based on the provided data (postId, title.rendered, and excerptRendered).

Right Articles are populated with creating and appending
const rightIds = [92, 95, 98];

```
// Fetch the post data from the API endpoint
```

```
fetch(
```

```
`https://excitable-breath.flywheelsites.com/wp-json/wp/v2/posts/${rightId}`
```

Here the Ahref goes to the article fetched for the option to comment on articles and create discussions.

Every Article holder has a hover Effect so it highlights to be more eyecatching.

above the middle section there is a button to "View All Articles"

this takes the user to section of all articles where their filtered with a map

```
const categoryMap = {
```

```
35: "technology",
```

```
30: "technology",
```

```
67: "technology",
```

```
65: "technology",
```

```
63: "technology",
```

```
61: "technology",
```

```
59: "technology",
```

```
84: "webDesign",
```

```
95: "webDesign",
```

```
98: "webDesign",
```

```
92: "webDesign",
```

```
69: "webDesign",
```

```
71: "trending",
```

```
57: "trending",
```

```
53: "trending",
```

```
55: "Trending",
```

```
33: "Trending",
```

```
24: "Trending",  
};
```

With this method the user can filter articles into categories and also add categories if needed. The Html structure here is

```
<select id="categoryFilter">
```

```
<option value="all">All</option>
```

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

```
<option value="webDesign">Web Design</option>
```

```
<option value="trending">Trending</option>
```

```
</select>
```

```
<input type="text" id="searchInput" placeholder="Search here..." />
```

via the navigation and footer at homePage User can go to Blog section. Here They are able to post a title and notes using airtable as a collector and post the information as given.

```
(async () => {  
  try {  
    const loadingDiv = document.getElementById("loading");  
    const contentDiv = document.getElementById("content");  
  
    // Show the loading indicator  
    loadingDiv.style.display = "block";  
    contentDiv.style.display = "none";  
  
    const response = await fetch(  
      "https://api.airtable.com/v0/apptXDjECOed3FGCl/Blog%20Posts?  
maxRecords=3&view=Grid%20view",  
      {  
        method: "GET",  
        headers: {  
          Authorization:  
            "Bearer  
patRYdlWCaONn6Gwy.50c762580e8bdba370c2abbe5b9fb2bbaed374ffe078a7ab  
6b78de93155a92fa",  
        },  
      }  
    );  
  
    if (response.ok) {  
      const result = await response.json();  
  
      // Remove the loading indicator and show the content  
      loadingDiv.style.display = "none";  
      contentDiv.style.display = "block";
```

```

// Create the innerHTML with the records data
let innerHTML = "";
result.records.forEach((record) => {
  const name = record.fields["Name"];
  const notes = record.fields["Notes"];
  innerHTML += `<div class="blog-wrapper">
<div class="blog-holder">
  <h2>${name}</h2>
  <br/>
<div class="text-con"
<p>${notes}</p>
<div class="btn-container">
  <button id="btnh1" class="btn">
    <i class="fa-regular fa-thumbs-up"></i>

  </button>

  <button id="btnh2" class="btn">
    <i class="fa-regular fa-thumbs-down"></i>

  </button>

</div>

</div>

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

```

The About page gives a short info about the company and how it operates.

In the contact page users can give their name, email and subject. Here The textarea is set to a minLength of 25 characters.

Name and Email have Regex for validation so form wont submit unless correct info is given.

```

const formEl = document.querySelector("#js-form");
const nameEl = document.querySelector("#js-name");
const emailEl = document.querySelector("#js-email");

formEl.addEventListener("submit", (event) => {
  // Stop the form from reloading the page
  event.preventDefault();

  const name = nameEl.value.trim();
  const email = emailEl.value.trim();

```

```

// Validate name
if (name === "") {
    alert("Please enter a name");
    return;
}

const minLengthRegex = /^[a-zA-Z]{2,}/;
if (!minLengthRegex.test(name)) {
    alert(
        "Name must be at least 2 characters long and must be letters only"
    );
    return;
}

// Validate email
if (email === "") {
    alert("Please enter an email");
    return;
}

const emailRegex =
    /[a-z0-9!#$%&'*/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*/=?^_`{|}~-]+)*@(?:[a-z0-9]
(?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?/;
if (!emailRegex.test(email)) {
    alert("Please enter a valid email address");
    return;
}

// Submit the form to an API
alert("Form submitted");
});

```