**Project Report**

**– Implementation of APIs in Website**

Inhaltsverzeichnis

[1. Introduction 2](#_Toc137734787)

[2. Project Scope and Objectives 2](#_Toc137734788)

[3. Project Implementation 2](#_Toc137734789)

[4. Results and Achievements 4](#_Toc137734790)

[5. Challenges and Lessons Learned 4](#_Toc137734791)

[6. Conclusion 4](#_Toc137734792)

[list of illustrations 4](#_Toc137734793)

# **1. Introduction**

This report summarizes the completion of a project involving the integration of different APIs into a website. The objective was to develop a website where users can retrieve jokes and facts by clicking on buttons. JavaScript, HTML, and CSS were used for implementation.

# **2. Project Scope and Objectives**

The project aimed to create a website that uses APIs to fetch and display jokes and facts. The scope included integrating three APIs, designing an interface with buttons, and implementing the logic to get the data from the APIs. The Steps involved API integration, designing the user interface, and implementing JavaScript for user interactions and displaying of the fetched data.

# **3. Project Implementation**

For the functionality we used buttons in HTML in which every button got his own id to be later accessed by the JavaScript code. The button itself was customized with the CSS Code “.btn & .btn: hover” which determines the size, color, margin and Hover effect. After receiving a message from the JavaScript Code, the message should be printed centred and blue as showcased in CSS Code “fact-text, joke-text {}” below.

***Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung***

Illustration 1: implementation of buttons and output text

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Multimedia, Display enthält.

Automatisch generierte Beschreibung

Illustration 2: implement output text design

Illustration 3: implement button design

Additionally, an API named “JokesOne” should be included where the user has the option to choose the category of the joke, he wants to receive himself. Implementing this part turned out to be a little harder but eventually we found the commands <select> and <option> and could successfully implement them.

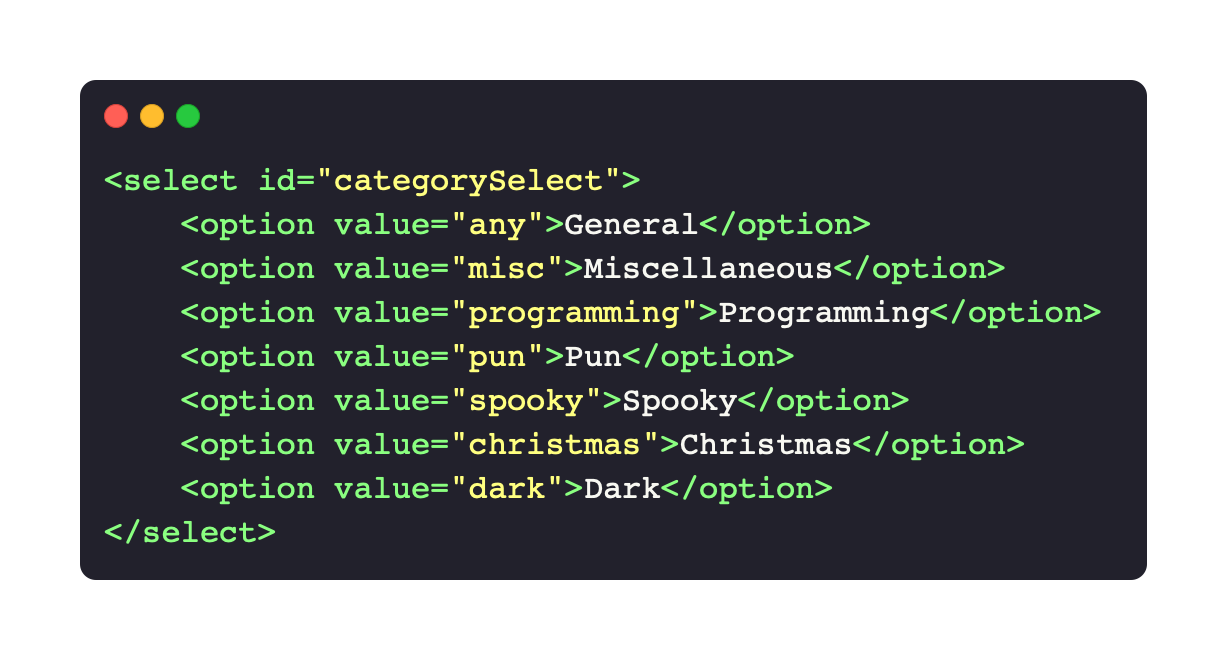
******

Illustration 4: implement categories, a button and output text

Implementing the functionality to fetch random facts from an API and updating the display accordingly was challenging during the project development. We used ChatGPT as a support to implement the error handling. It was hard, but after we got the fetchRandomFact method and the error handling it was much easier to finfish the code.

Ein Bild, das Text, Elektronik, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Illustration 5: implement javaScript code to receive fact from API

# **4. Results and Achievements**

The project successfully created a website where users can click on buttons to retrieve random jokes and facts from “chuckNorris API”, “JokeOne API”, and “useless facts API”. The website provides a fairly large range of content.

# **5. Challenges and Lessons Learned**

We learned how to include APIs in our programs, implement JavaScript itself on the website, using buttons and the <select> statement, work with error handling and data manipulation.

# **6. Conclusion**

The project was a very fun and teaching experience where we could learn lots of new things as already said in point four but also things like teamwork and organisation grew to be an important part.

**list of illustrations**

[Illustration 1: implementation of buttons and output text 2](#_Toc137735277)

[Illustration 2: implement output text design 2](https://d.docs.live.net/4b5ff72d082cdfdc/Dokumente/Projectreport.docx#_Toc137735278)

[Illustration 3: implement button design 2](https://d.docs.live.net/4b5ff72d082cdfdc/Dokumente/Projectreport.docx#_Toc137735279)

[Illustration 4: implement categories, a button and output text 3](#_Toc137735280)

[Illustration 5: implement javaScript code to receive fact from API 3](#_Toc137735281)