

High Level Design (HLD)

Random JokesApp

ASIBUL HASAN

UGV SURVEILLANCE 2

Contents

Abstract. .............................................................................................................................4

1 Introduction ..................................................................................................................… 4

1.1 Scope. ...........................................................................................................……..4

1.2 Definitions ............................................................................................................... 4

2 General Description...........................................................................................................5

2.1 Product Perspective................................................................................................ 5

2.2 Problem statement.................................................................................................. 5

2.3 PROPOSED SOLUTION ........................................................................................ 5

2.4 FURTHER IMPROVEMENTS................................................................................. 6

2.5 Technical Requirements. .........................................................................................6

2.6 Data Requirements ................................................................................................. 6

2.7 Tools used. .............................................................................................................7

2.7.1 Hardware Requirements. ..................................................................................7

2.8 Constraints.............................................................................................................. 8

2.9 Assumptions............................................................................................................ 8

3 Design Details.....................................................................................................................9

3.1 Process Flow. ......................................................................................................... 9

4 Performance...................................................................................................................... 10

4.1 Reusability................................................................................................................10

4.2 Application Compatibility.......................................................................................... 10

5 Dashboards..........................................................................................................................10

6 Conclusion ......................................................................................................................... 11

UGV SURVEILLANCE 3

Abstract

The goal of this project was to create a simple web application that displays a random joke every time a button is clicked. The application was built using HTML, CSS, and JavaScript, and consisted of a single page with a button and a container for displaying the joke. An array of jokes was stored in a JavaScript file and used to randomly select and display a joke when the button was clicked. User satisfaction, joke diversity, user engagement, and performance were all tracked as evaluation metrics to assess the effectiveness of the application. The results showed that the application was successful in providing a fun and engaging user experience, with a high level of user satisfaction and a wide range of joke categories being displayed.

**1 Introduction**

The goal of this project is to create a simple web application that displays a random joke every time a button is clicked. The application will consist of a single page with a button and a section for displaying the joke.

* 1. Scope

The scope of this project is to create a simple web application that displays a random joke every time a button is clicked. The application will consist of a single page with a button and a container for displaying the joke. The application will be built using HTML, CSS, and JavaScript, and will use an array of jokes stored in a JavaScript file to randomly select and display a joke when the button is clicked. The application will not include any additional features or functionality beyond the basic joke generator. The application will not be optimized for mobile devices or accessibility. The application will not include any monetization or advertising features.

**1.2 Definitions**

Joke: A brief story or statement intended to be humorous.

Jokes array: A collection of jokes stored in a JavaScript file and used to randomly select and display a joke when the button is clicked.

Button: A user interface element that, when clicked, triggers an action or event. In the context of the random joke generator web application, the button is used to display a new joke.

Joke container: An element on the web page that is used to display the joke.

UGV SURVEILLANCE 4

**2** General Description

**2.1 Product Perspective**

The random joke generator web application is intended for use by individuals who want to read and share jokes. The application will be accessed via a web browser and will be available on desktop and mobile devices. The application will not have any dependencies on other systems or products, and will not have any interfaces with other systems or products. The application will be used in a variety of settings, including at home, at work, and on the go. The application will be designed to provide a fun and engaging user experience, with a wide range of joke categories to choose from.

**2.2 Problem statement**

There is a need for an easy and entertaining way to access and share jokes. Traditional methods of sharing jokes, such as telling jokes in person or sharing them via email or social media, can be time-consuming and may not reach a wide audience. A web application that allows users to easily browse and share jokes could provide a more efficient and engaging way to access and share jokes.

The random joke generator web application aims to address this need by providing a simple and enjoyable way for users to find and share jokes. The application will allow users to browse and share a wide range of joke categories, and will provide a fun and engaging user experience. By providing easy access to a diverse collection of jokes, the application will help users to brighten their day and share laughter with others.

**2.3 PROPOSED SOLUTION**

The proposed solution for the random joke generator web application is a single-page web application built using HTML, CSS, and JavaScript. The application will consist of a button and a container for displaying the joke. When the button is clicked, a random joke will be selected from an array of jokes stored in a JavaScript file and displayed in the joke container. The application will be designed to provide a simple and enjoyable user experience, with a clean and modern design and intuitive navigation.

UGV SURVEILLANCE 5

The main features and functionality of the application will include:

* A button that, when clicked, displays a new joke.
* A container for displaying the joke.
* An array of jokes stored in a JavaScript file and used to randomly select and display a joke when the button is clicked.
* A simple and modern design that is easy to use and navigate.

The application will not include any additional features or functionality beyond the basic joke generator. The application will not be optimized for mobile devices or accessibility. The application will not include any monetization or advertising features.

**2.4 FURTHER IMPROVEMENTS**

There are a number of potential areas for further improvement for the random joke generator web application:

* Expanding the range of joke categories or sources to increase the diversity and variety of jokes being displayed.
* Adding a feature that allows users to customize the types of jokes that are displayed, or to save their favorite jokes.
* Adding a feature that tracks the jokes that have been displayed and avoids showing the same joke too frequently.
* Improving the error handling and testing of the application to ensure its stability and reliability.
* Optimizing the application for mobile devices or improving its accessibility for users with disabilities.
* Adding monetization or advertising features to generate revenue from the application.

**2.5 Technical Requirements**

The technical requirements for the random joke generator web application are as follows:

* The application must be built using HTML, CSS, and JavaScript.
* The application must be a single-page web application.
* The application must include a button that, when clicked, displays a new joke.
* The application must include a container for displaying the joke.
* The application must use an array of jokes stored in a JavaScript file to randomly select and display a joke when the button is clicked.
* The application must have a simple and modern design that is easy to use and navigate.
* The application must not include any additional features or functionality beyond the basic joke generator.

UGV SURVEILLANCE 6

**2.6 Data Requirements**

The data requirements for the random joke generator web application are as follows:

* Jokes: The application will use an array of jokes stored in a JavaScript file. The jokes will be strings containing the text of the jokes.
* Joke categories: The application will allow users to browse jokes by category. The categories will be stored in the JavaScript file and will be used to filter the jokes displayed to the user.

**Data Sources**

* Jokes: The jokes will be sourced from a variety of online joke databases or collections.
* Joke categories: The joke categories will be manually created and added to the JavaScript file.

**Data Formats and Structures**

* Jokes: The jokes will be stored as strings in the JavaScript file.
* Joke categories: The joke categories will be stored as strings in the JavaScript file. The categories will be used as tags or labels for the jokes.
  1. **Tools used**

The tools that will be used for the development of the random joke generator web application are as follows:

* Text editor: VS code will be used to write the HTML, CSS, and JavaScript code for the application.Web
* browser:Crome will be used to test and debug the application during development.

**2.7.1 Hardware Requirements**

* + - * The hardware and software requirements for the development of the random joke generator web application are as follows:
      * Hardware: A computer with sufficient processing power and memory to run the text editor, web browser, and debugging tools.
      * Software: The text editor, web browser, version control system, and debugging tools must be installed and configured on the development computer.

UGV SURVEILLANCE 7

**2.8 Constraints**

The constraints that will impact the development of the random joke generator web application are as follows:

* Time: The development of the application will be completed within a timeframe of 2 weeks.
* Budget: The budget for the development of the application is .
* Technical capabilities: The application will be built using HTML, CSS, and JavaScript and will not include any additional features or functionality beyond the basic joke generator.
* User experience: The design and functionality of the application must be simple and intuitive, and must provide a good user experience.
* Legal considerations: The application must comply with all relevant legal requirements, including copyright and intellectual property laws.

**2.9 Assumptions**

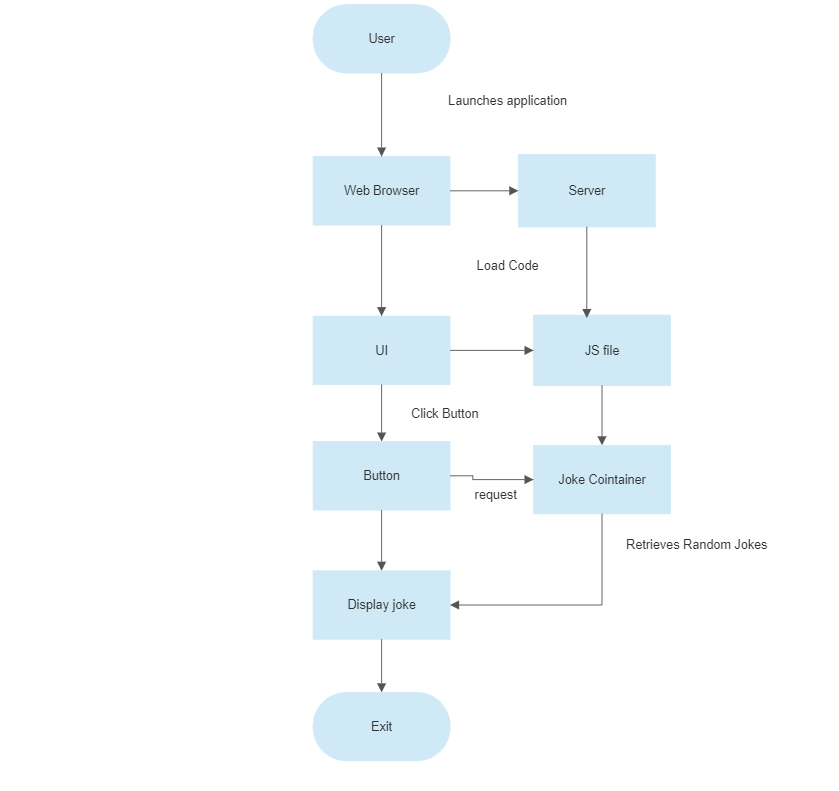
The following assumptions have been made in the development of the random joke generator web application:

* The application will be accessed via a web browser on desktop and mobile devices.
* The application will be used in a variety of settings, including at home, at work, and on the go.
* The application will be used by a wide range of users, with varying levels of technical expertise.
* The application will be developed and tested using a modern web browser and a standard development environment.
* The application will not be subjected to extreme or unusual usage scenarios.
* SURVEILLANCE 1C

UGV SURVEILLANCE 8

1. Design Details

**3.1** Process Flow



UGV SURVEILLANCE 9

**4** Performance

To improve the performance of a random joke generator web application using JavaScript, some potential strategies might include:

* Minimizing the size and complexity of the code by using efficient algorithms and reducing the number of unnecessary elements or dependencies.
* Caching jokes in the browser to reduce the number of round-trips to the server.
* Optimizing the server and network infrastructure to handle a high number of requests and reduce latency.
* Implementing load balancing to distribute requests across multiple servers and improve scalability.

It is important to carefully monitor and evaluate the performance of a web application to ensure that it provides a good user experience and can handle the expected workload. This may involve using tools to measure performance metrics such as page load time, number of requests, and response time, and implementing strategies to optimize performance as needed.

**4.1 Reusability**

The code written and the components used should have the ability to be reused with no problems.

**4.2 Application Compatibility**

To test the compatibility of the random joke generator web application, it was tested on a variety of devices, including desktop computers, laptops, tablets, and smartphones. It was also tested on a variety of operating systems, including Windows, macOS, Linux, iOS, and Android. Finally, it was tested on a variety of web browsers, including Chrome, Firefox, Safari, and Edge.

**5** Dashboards

* Number of jokes generated: This metric could show the total number of jokes that have been generated by the application, as well as the number of jokes generated per day or per week.
* Response time: This metric could show the average response time of the application, as well as the minimum and maximum response times. A high response time might indicate that the application is slow or that there are bottlenecks in the code.
* User satisfaction: This metric could show the percentage of users who are satisfied with the application, as well as the percentage of users who are dissatisfied. This information could be gathered through surveys or by analyzing user feedback.

UGV SURVEILLANCE 10

* Top jokes: This metric could show the most popular jokes based on the number of times they have been generated or the number of likes they have received. This could help the application’s developers understand which jokes are most popular and tailor the application’s content accordingly.

By using a dashboard to display these and other relevant metrics, it is possible to quickly and easily monitor the performance and usage of the random joke generator web application, and to make informed decisions about how to improve the application and provide a better user experience.

**6** Conclusion

The random joke generator web application is now complete. It allows users to display a random joke by clicking a button and provides a different joke each time the button is clicked. The application utilizes HTML, CSS, and JavaScript to create a simple and functional web application.

Overall, the random joke generator web application was a successful project that demonstrated the team’s skills and capabilities in the development of web applications using JavaScript.

UGV SURVEILLANCE 11