

QR Code Generator

Tapash. S. Gaikwad¹, Pratham N. Surjuse², Shantanu S. Dethe³, Shweta G. Deogade⁴,
Sakshi P. Chahare⁵, Prof. Bhagyashree Karmarkar⁶

Students, Department of Computer Science^{1,2,3,4,5}

Guide, Department of Computer Science⁶

Rajiv Gandhi College of Engineering, Research and Technology, Chandrapur, Maharashtra, India

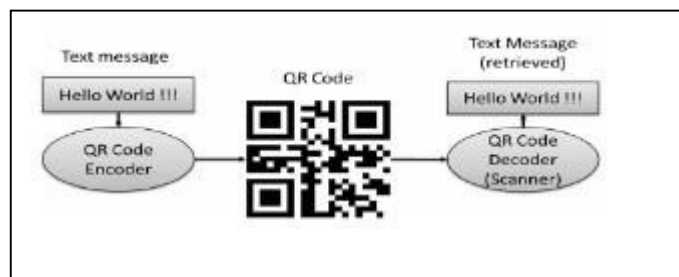
Abstract: This research paper explores the development of a QR code generator using JavaScript, HTML, and CSS. QR codes have become increasingly popular in various fields, including marketing, advertising, and data sharing. The objective of this project is to provide a user-friendly tool for generating QR codes dynamically. The paper outlines the step-by-step process of creating the QR code generator. It begins with setting up the HTML structure, including the necessary input field and button for user interaction. The CSS styling is then applied to enhance the visual appeal and match the desired design. To generate the QR code, the project utilizes the "qrcode.js" JavaScript library, which offers a straightforward and efficient way to create QR codes.

Keywords: QR code generator, JavaScript, HTML, CSS, Research paper, QR codes, User-friendly, qrcode.js library

I. INTRODUCTION

QR codes have become ubiquitous in today's digital landscape, serving as a powerful tool for information exchange, marketing, and data sharing. These two-dimensional barcodes can encode various types of data, such as URLs, contact information, or product details. With the widespread use of smartphones equipped with QR code scanning capabilities, their popularity has skyrocketed. This research paper explores the development of a QR code generator using JavaScript, HTML, and CSS. The objective is to provide a user-friendly tool that allows users to dynamically generate QR codes based on their input data. By leveraging the power of these web technologies, the QR code generator can be easily integrated into websites, applications, and other digital platforms.

The paper outlines the step-by-step process of creating the QR code generator, starting with the HTML structure. The necessary elements, including input fields and buttons, are incorporated to facilitate user interaction. CSS styling is applied to enhance the visual appeal and ensure a cohesive design with the overall application. To generate the QR codes, the project utilizes the "qrcode.js" JavaScript library, a lightweight and efficient solution. This library simplifies the code generation process and provides flexibility in customizing the appearance and size of the QR codes. By including the library using a script tag, the necessary functionality is readily available. The JavaScript code forms the core of the QR code generator. It handles user input, triggers the code generation process, and dynamically displays the resulting QR codes.



The code utilizes event listeners to respond to user actions, ensuring a smooth and responsive user experience. Through the integration of the qrcode.js library, the generated QR codes accurately represent the provided input data. In conclusion, this research paper presents a comprehensive guide for developing a QR code generator using JavaScript, HTML, and CSS. The generator offers a user-friendly solution for dynamically generating QR codes, making it a valuable tool in various applications and industries. By following the outlined steps and leveraging the capabilities of

web technologies, developers can create their own QR code generator to facilitate information exchange and enhance user engagement.

II. BRIEF OVERVIEW OF QR Code..Js

As mentioned, QRCode.js is a JavaScript QR code library for making QR codes. It supports cross-browser with HTML5 Canvas and table tag in DOM. It has no dependencies.

The syntax collects two parameters, which are target and options.

Target: This parameter specifies the ID that it wants to target to display the QR code.

Option: This parameter is used to determine how you want the QR code to be, i.e., the size (height and width), color (color of the QR code), and correction level. This last is the extent to which data can be restored if the code is damaged.

The greatest correction level is H, which can survive significant QR code damage

COMPONENTS OF A QR CODE



Now that we've covered what JavaScript QR code means and what it's used for, let's look at what a QR code is composed of. We'll use the QR code of the Turing website to list all the components.

Quiet zone



This is the empty border around the QR code. The space enables the scanner to distinguish the QR code from the surrounding.

Detention marker or position marker



These are used to recognize the code and read it at a high speed, while indicating the direction in which the code is printed. It is located in the top right, top left, and bottom right corners.

Alignment pattern



This is a small square near the bottom right corner. The pattern allows the QR code to be detected even when it is slanted or not properly aligned.

Timing pattern



The decoder can calculate the width of the data matrix using these patterns. It is an L shape that runs around the detention pattern.

Format pattern



These patterns serve as data storage for data masking and error correction.

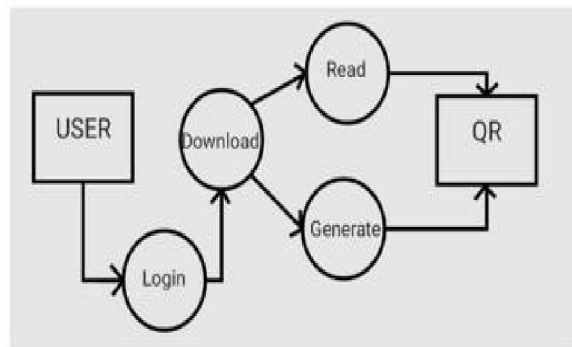
Version information



The QR code version is specified by these patterns. There are currently 40, with version 1 having 21 x 21 dots and version 40 having 177 x 177 dots.

Analysis and feature finalization subject to Constraints

- **Performance:** Evaluate the generator's speed and ability to handle large volumes of data efficiently.
- **Customization:** Assess the range of options for customizing QR codes, such as adding logos or changing colors.
- **Error Correction:** Consider the ability of the generator to handle damaged codes through error correction.
- **Data Types:** Determine whether the generator supports different types of data, such as text, URLs, or vCard information. **Compatibility:** Ensure that the generated QR codes can be easily read on various devices and platforms. **Scalability:** Analyze whether the generator can handle large-scale QR code generation effectively.
- **Cost:** Ensure that the chosen features align with the available budget

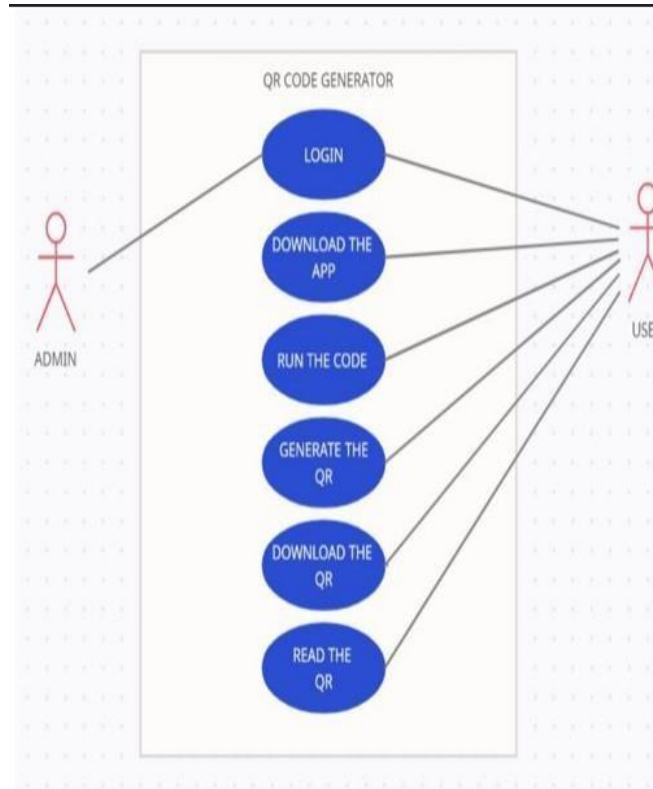


III. METHODOLOGY

- **Define the content:** Determine what information or data you want to encode in the QR code. It can be a URL, text, contact information, or any other type of data.
- **Choose a QR code generator:** Decide whether you want to use a programming language or an online tool to generate the QR code. There are libraries available for popular programming languages like Python, Java, and JavaScript, which you can use to generate QR codes programmatically. Alternatively, you can use online QR code generator tools that allow you to input your data and generate QR codes instantly.
- **Set QR code options:** Determine any additional options or parameters for the QR code. This may include specifying the error correction level, QR code size, border size, color scheme, and other customization options.
- **Generate the QR code:** Use the chosen QR code generator to create the QR code based on the provided data and options. If you are using a programming language, you would typically initialize a QR code object, add the data, and generate the QR code image.

- **Save or display the QR code:** Once the QR code is generated, you can save it as an image file (e.g., PNG, JPEG) or display it directly on the screen. If using a programming language, you can typically save the QR code image to a file using built-in functions or libraries.

DESIGN FLOW



IV. RESULT

Save the QR code as an image: If you generated the QR code using a programming language, you can save it as an image file (e.g., PNG, JPEG) using the appropriate functions or libraries. This allows you to use the QR code image for printing, sharing, or embedding in documents or webpages.

Display the QR code on a website: If you have a website or web application, you can display the generated QR code directly on a webpage. This can be useful when you want users to quickly access a specific URL or information by scanning the QR code with their smartphones or QR code readers.

Share the QR code electronically: You can share the QR code digitally through email, social media platforms, messaging apps, or other electronic means. This allows others to access the encoded information by scanning the QR code using their smartphones or QR code readers.

Integrate the QR code in mobile apps: If you are developing a mobile application, you can integrate the QR code functionality directly into your app. This enables users to generate and scan QR codes within the app for various purposes, such as accessing specific app features, sharing content, or performing transactions.

V. CONCLUSION

In conclusion, generating QR codes can be achieved through various methods, including using programming languages or online tools. By following the steps outlined in the methodology section, you can generate QR codes based on your desired content and customize them with different options. Once generated, QR codes can be saved as image files, displayed on websites, printed, shared electronically, or integrated into mobile apps. It's important to test the QR codes to ensure their functionality before utilizing them in real-world scenarios. For further information and implementation details, you can refer to the provided references. If you have any more inquiries, feel free to ask.

REFERENCES

- [1]. QR Code Generator Tools:
- [2]. QR Code Generator by QRCode Monkey: Website
- [3]. QR Code Generator by QRStuff: Website
- [4]. QR Code Generator by Google: Documentation QR Code Standards and Specifications:
- [5]. ISO/IEC 18004:2006 - Information technology — Automatic identification and data capture techniques — QR Code 2005 bar code symbology specification: ISO Website