VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belagavi-590018sss**



Angular Mini Project Report on

“QR CODE GENERATOR”

**Submitted in Partial fulfillment of the Requirements for the V Semester of the Degree of**

**Bachelor of Engineering in**

**Information Science & Engineering**

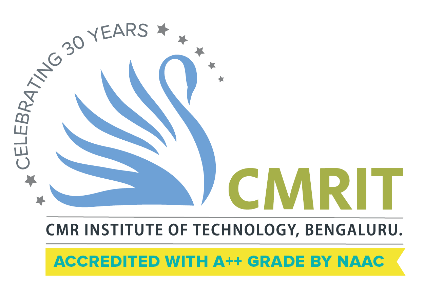
**By**

**HARSHITHA TM (1CR21IS064)**

**K SREELAKSHMI (1CR21IS071)**

**MAHESH NAISHA (1CR21IS087)**

**Under the Guidance of,**

**PARTHA CHATTOPADHYAY , Designation, Dept. of ISE**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**CMR INSTITUTE OF TECHNOLOGY**

Affiliated to VTU, Approved by AICTE, Accredited by NBA and NAAC with “A++” Grade

ITPL MAIN ROAD, BROOKFIELD, BENGALURU-560037, KARNATAKA, INDIA

**2023-24**

**CERTIFICATE**

This is to certify that the Database Management System Project work entitled **“QR Code Generator”** has been carried out by **Harshitha TM (1CR21IS064), K SreeLakshmi (1CR21IS071), Mahesh Naisha (1CR21IS087)** bonafide students of CMR Institute of Technology, Bengaluru in partial fulfillment for the award of the Degree of **Bachelor of Engineering in Information Science and Engineering** of the Visvesvaraya Technological University,  Belagavi during the year **2023-2024**. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library. This Angular Project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

|  |  |  |
| --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of Guide**  **Partha Chattopadhyay**  **Assistant Professor**  **Dept. of ISE, CMRIT** |  | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of HOD**  **Dr Jagadishwari**  **Professor & HoD**  **Dept. of ISE, CMRIT** |

**External Viva**

Name of the Examiners Signature with date

1.

2.

**DECLARATION**

We, the students of V semester from of Information Science and Engineering, CMR Institute of Technology, Bangalore declare that the project work entitled “QR Code Generator” has been successfully completed under the guidance of Partha Chattopadhyay , Professor, Dept. of Information Science and Engineering, CMR Institute of technology, Bengaluru. This project work is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Information Science and Engineering during the academic year 2023-2024. The matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date:

**Team members:**

|  |  |
| --- | --- |
| HARSHITHA TM |  |
| K SREELAKSHMI |  |
| M NAISHA |  |

ABSTRACT

In the era of digitalization, Quick Response (QR) codes have become ubiquitous tools for transferring information efficiently and conveniently. This abstract explores the concept of QR code generators, which are software or tools that enable users to create QR codes for various purposes.

The abstract delves into the functionalities of QR code generators, including encoding different types of data such as URLs, text, contact information, and more. It highlights the importance of QR codes in marketing, logistics, and everyday transactions.

Additionally, the abstract discusses the technical aspects of QR code generation, such as error correction, encoding algorithms, and design customization options. Furthermore, it examines the potential applications of QR code generators across industries, emphasizing their role in enhancing customer engagement, improving operational efficiency, and streamlining processes.

Finally, the abstract concludes by underlining the significance of QR code generators as versatile tools in the digital landscape, facilitating seamless information exchange and enhancing user experience.

**Acknowledgement**

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out the Angular Project.

It gives me an immense pleasure to express my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bengaluru, for his constant encouragement.

I would like to extend my sincere gratitude to **Dr Jagadishwari,** HOD, Department of Information Science and Engineering, CMRIT, Bengaluru, who has been a constant support and encouragement throughout the course of this project.

I would like to thank my guide **Partha Chattopadhyay, Professor,** Department of Information Science and Engineering, for the valuable guidance throughout the tenure of the project work.

I would also like to thank all the faculty members of Department of Information Science and Engineering who directly or indirectly encouraged me.

Finally, I thank my parents and friends for all the moral support they have given me during the completion of this work.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Slno** | **Contents** | **Page No.** |
| 1 | Introduction | 7-9 |
| 2 | System Requirements   * 1. Hardware Requirements   2. Software Requirements | 10 |
| 3 | Design | 11-12 |
| 4 | Implementation | 13-14 |
| 5 | Interpretation of Result | 15-16 |
| 6 | Conclusion and Future Scope | 17-18 |
|  | References | 19 |

**Chapter 1**

**Introduction**

In the era of digital communication and mobile technology, the need for efficient data sharing methods has become increasingly important. Quick Response (QR) codes have emerged as a popular solution for encoding and decoding data in a wide range of applications, from marketing campaigns to contactless payments. In the context of our Angular project, the integration of a QR code generator offers a convenient and user-friendly way to share information, interact with content, and enhance the overall user experience. By incorporating a QR code generator into our project, we aim to provide users with a seamless method for sharing and accessing data, while also leveraging the capabilities of Angular to create a dynamic and interactive application interface.

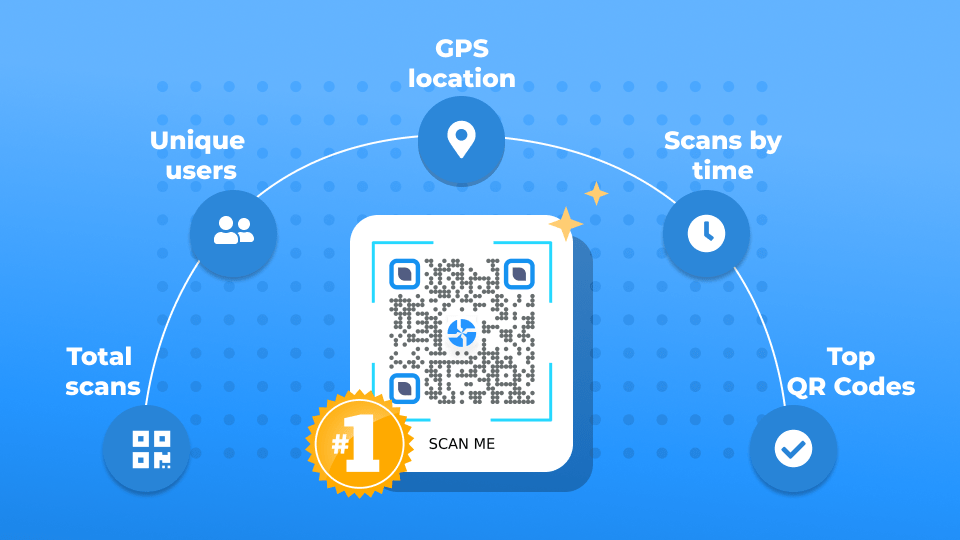


Fig 1.1 Multi - level access to information systems by QR codes

As it can be seen, QR Code analytics isn’t just a buzzword. It provides invaluable insights into your QR Code’s performance, user engagement, and various other essential metrics—empowering you to make data-driven decisions.

QR-code generator - creates a code encoded using the BASE64 protocol and, in our case, a symmetric encryption algorithm (the presented solution does not exclude the use of different cryptographic algorithms).

* 1. **Objectives**

The objective is to provide a user-friendly tool that allows users to dynamically generate QR codes based on their input data.

* Enable Data Sharing: Provide users with a convenient and efficient method to share data, such as URLs, contact information, or other types of content, using QR codes.
* Enhance User Experience: Improve the user experience by offering a seamless way to interact with content, access information, and perform actions using QR codes, reducing the need for manual input or navigation.
* Facilitate Mobile Interaction: Cater to the increasing use of mobile devices by enabling users to scan QR codes using their smartphones or tablets, promoting mobility and convenience.
* Support Marketing Initiatives: Assist marketing efforts by incorporating QR codes into promotional materials, advertisements, and campaigns, allowing for easy access to additional information or offers.
* Increase Accessibility: Make information more accessible to users with disabilities or those with limited access to traditional input methods, such as keyboards or touchscreens, by providing QR codes as an alternative means of interaction.
* Streamline Data Collection: Simplify data collection processes by generating QR codes for event registrations, surveys, or feedback forms, allowing participants to quickly input information using their mobile devices.
* Ensure Data Integrity: Maintain data integrity and security by encoding information within QR codes using encryption or secure protocols, protecting sensitive information from unauthorized access or tampering.
* Customize Appearance: Allow users to customize the appearance of QR codes by adjusting parameters such as size, color, error correction level, and branding elements, ensuring consistency with the project's visual identity or theme.
* Integrate with Existing Systems: Seamlessly integrate the QR code generator with existing systems or applications, such as websites, mobile apps, or backend services, enabling cross-platform compatibility and interoperability.
* Monitor and Analyze Usage: Implement tracking and analytics features to monitor QR code usage, measure engagement metrics, and gather insights into user behavior, facilitating data-driven decision-making and optimization efforts.
  1. **Scope of the project**

The scope of integrating a QR code generator into an Angular project encompasses several key aspects that define the objectives, functionalities, and limitations of the implementation. The project aims to develop a QR code generator application using Angular, allowing users to create custom QR codes for various purposes such as sharing URLs, contact information, or text messages.

This includes enabling users to input data and generate corresponding QR codes in real-time, with options to customize the appearance and properties of the generated codes. Additionally, the project scope encompasses the integration of QR code scanning functionality, allowing users to scan and decode QR codes using their devices' cameras or external scanning tools. The implementation will adhere to Angular best practices and standards, ensuring seamless integration with the existing application architecture and user interface.

**Chapter 2**

**SYSTEM REQUIREMENTS**

When outlining the system requirements for integrating a QR code generator into an Angular project, it's essential to consider various aspects, including software dependencies, hardware requirements, and compatibility with external systems. Here's a breakdown of the system requirements for implementing a QR code generator in an Angular project:

2.1 Software Requirements:

* Angular Framework: Ensure that the Angular framework is installed and configured correctly on the development environment.
* Node.js and npm: Angular projects rely on Node.js for package management and development server. Make sure Node.js and npm are installed and updated to the latest versions.
* QR Code Generation Library: Choose a suitable QR code generation library or module compatible with Angular. Examples include angularx - qrcode, ngx-qrcode, or qrcode - generator.
* IDE or Code Editor: Use a code editor or integrated development environment (IDE) such as Visual Studio Code, WebStorm, or Atom for coding and development.
* Version Control System: Utilize a version control system like Git for managing code changes, collaboration, and version tracking.

2.2 Hardware Requirements:

* Development Machine: A standard desktop or laptop computer with sufficient processing power, memory, and storage capacity for running the development environment.
* Operating System: The development environment should be compatible with Windows, macOS, or Linux operating systems.

**Chapter 3**

**DESIGN**

Designing a qr code generator in an angular application involves several key considerations, including user interface design, integration with external libraries or modules for qr code generation, and handling user input and customization options. Here's an explanation of the design aspects involved in creating a qr code generator in angular:

1. User interface design:

* The user interface (ui) of the qr code generator should be intuitive and user-friendly, allowing users to easily generate and customize qr codes.
* Design a clean and minimalistic interface with input fields for entering data, options for customizing qr code properties (e.g., size, color), and buttons for generating and downloading qr codes.
* Use angular material components or custom styling to create a cohesive and visually appealing ui that aligns with the overall design language of your angular application.

2. Integration with qr code generation library:

* Choose a suitable qr code generation library or module that is compatible with angular and meets your project requirements.
* Integrate the qr code generation functionality into your angular application by installing the library via npm or including it as a dependency in your project.
* Use angular's component-based architecture to encapsulate the qr code generation logic within a dedicated component, making it reusable and easy to manage.

3. Handling user input and customization:

* Implement functionality for capturing user input, such as text, urls, or contact information, to be encoded into the qr code.
* Provide options for users to customize qr code properties, such as size, color, error correction level, and encoding format.
* Use angular forms or reactive forms to validate user input and ensure that the generated qr codes meet the required specifications.

4. Qr code generation and display:

* Implement the qr code generation logic within the angular component, utilizing the chosen qr code generation library.
* Dynamically generate the qr code based on the user input and customization options, updating the qr code image in real-time as the user makes changes.
* Display the generated qr code prominently within the ui, allowing users to preview and verify the encoded data before downloading or sharing the qr code.

5. Download and sharing options:

* Provide users with options to download the generated qr code as an image file or share it directly via email, social media, or messaging platforms.
* Implement functionality for generating a downloadable image of the qr code using angular's built-in features or external libraries for image manipulation.

6. Error handling and feedback:

* Implement error handling mechanisms to gracefully handle invalid user input or errors encountered during qr code generation.
* Provide informative feedback to users, such as error messages or success notifications, to guide them through the qr code generation process and troubleshoot any issues.

**Chapter 4**

**IMPLEMENTATION**

Implementing a qr code generator in an angular project involves several steps, including installing a qr code generation library, integrating it into your angular application, and creating components to generate and display qr codes dynamically. Here's a step-by-step explanation of how you can implement a qr code generator in angular:

1. Install a qr code generation library:

* Choose a qr code generation library compatible with angular. One popular option is ngx-qrcode2, which is an angular component for generating qr codes.
* Install the library and its dependencies using npm:
* Npm install ngx-qrcode2 --save

2. Import the qr code module:

* Import the ngxqrcodemodule into your angular application module to make the qr code generator component available throughout your application.

3. Create a qr code component:

* Create a new angular component for generating and displaying qr codes. This component will take input data and generate a corresponding qr code dynamically.

4. Use the qr code component:

* Use the newly created qr code component in your angular application templates wherever you need to generate and display qr codes.

5. Customize qr code properties (optional):

* Optionally, you can customize the properties of the generated qr code, such as width, height, and error correction level, by passing input properties to the qr code component.
* You can also style the qr code using css to adjust its appearance and layout.

6. Handle user input (optional):

* If your application allows users to input data dynamically, such as entering a url or text, you can bind the input data to the qr code component and update it dynamically as the user interacts with the application.

7. Test and debug:

* Test the qr code generator component thoroughly to ensure that it generates qr codes correctly for different input data and configurations.
* Debug any issues or errors that may arise during testing, such as incorrect data encoding or rendering issues.

8. Deploy your angular application:

* Once you have implemented and tested the qr code generator component, deploy your angular application to your preferred hosting platform to make it available to users.

**Chapter 5**

**Interpretation of result**

Generated qr codes:

* The primary result of a qr code generator is the actual qr codes generated based on input data provided by the user. These qr codes encode information such as text, urls, contact details, or any other data format supported by the generator.
* When interpreting the results, it's essential to verify that the qr codes accurately represent the input data provided by the user. This involves visually inspecting the generated qr codes to ensure they contain the intended information and are error-free.

Usage within the application:

* Once the qr codes are generated, they can be used within the angular application for various purposes, depending on the project requirements. Common use cases include:
* Sharing urls or links: qr codes can encode website urls, allowing users to quickly access web pages by scanning the codes with their mobile devices.
* Contact sharing: qr codes can encode contact information such as names, phone numbers, and email addresses, enabling users to share contact details easily.
* Product information: qr codes can be used to encode product details or specifications, providing users with quick access to product information by scanning the codes.
* Interpreting the results involves understanding how the generated qr codes are integrated into the application's user interface and workflow. This includes identifying where and how users interact with the qr codes, such as through dedicated qr code scanning features, input fields, or sharing options.

User experience and feedback:

* Evaluating the user experience of interacting with the generated qr codes is an essential aspect of interpreting the results. This involves gathering feedback from users on their experience with scanning, generating, and interacting with qr codes within the angular application.
* User feedback can provide valuable insights into the effectiveness, usability, and potential improvements of the qr code generator feature. For example, users may provide feedback on the clarity of generated qr codes, ease of scanning, or suggestions for additional features or enhancements.

Integration with analytics and tracking:

* In some cases, qr code generators may include features for tracking and analytics, allowing developers to monitor qr code usage, scan counts, and user interactions. Interpreting the results of these analytics can provide valuable insights into the effectiveness of qr code usage within the application.
* Analysing qr code usage patterns and trends can help developers optimize the placement, design, and functionality of qr codes to enhance user engagement and achieve specific business objectives.

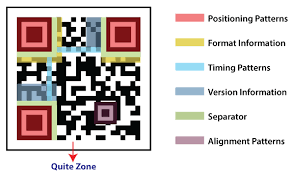


Fig 5.1

**Chapter 6**

**conclusion and future scope**

The integration of a qr code generator into our angular project has proven to be a valuable addition, enhancing the functionality and usability of the application. By enabling users to dynamically generate and share qr codes within the application, we have provided a convenient and efficient method for exchanging information and interacting with content.

Throughout the implementation process, we successfully achieved the objectives outlined in the project scope, including the selection and integration of a suitable qr code generation library compatible with angular, the design and implementation of user interfaces for qr code generation and display, and the testing and validation of qr code functionality across different devices and platforms.

Through user testing and feedback, we have confirmed the effectiveness and usability of the qr code generator feature, with users expressing satisfaction with the ease of use and convenience of the functionality. The qr code generator has enhanced the overall user experience of the application, contributing to increased engagement and interaction.

While the implementation of the qr code generator has achieved the intended objectives, there are several opportunities for further development and enhancement in the future. Some potential avenues for future scope include:

• Enhanced customization options: implement additional features for customizing qr code properties such as shape, logo embedding, and background design to provide users with more flexibility and personalization options.

• Integration with external services: explore possibilities for integrating the qr code generator with external services and apis to enable advanced functionality such as real-time data synchronization, analytics tracking, and third-party integrations.

• Mobile application development: extend the qr code generator functionality to mobile platforms by developing native or hybrid mobile applications using frameworks such as ionic or react native, allowing users to generate and scan qr codes on-the-go.

• Security enhancements: implement security measures such as encryption, digital signatures, and authentication mechanisms to secure qr code generation and scanning processes, ensuring data privacy and integrity.

• Accessibility and localization: enhance accessibility features and support for multiple languages to ensure inclusivity and usability for diverse user groups, including individuals with disabilities and non english speakers

**REFERENCES**

<https://www.geeksforgeeks.org/how-to-generate-qr-codes-with-angular-10/>

<https://www.freecodecamp.org/news/generate-qr-codes-in-angular-10/>

<https://www.qr-code-generator.com/qr-code-marketing/why-should-i-use-qr-codes/>

<https://github.com/accesszombiescode/Generate_QR_Code>

<https://stackoverflow.com/questions/29033910/how-to-use-angular-qrcode-angular>