

Ramanand Arya D.A.V.College, Automonous

(Affiliated to University of Mumbai)

MUMBAI, 400042



DEPARTMENT OF INFORMATION TECHNOLOGY

A Project Report on

QR CODE GENERATOR

Submitted in partial fulfillment of the requirements for the award the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

(Academic Year 2023-2024)

By

SUMIT DASHRATH GUPTA

Seat No: 556

Under the esteemed guidance of

Mrs. Amruta Joshi

Assistant Professor

CERTIFICATE

This is to certify that the project entitled, “**QR CODE GENERATOR WEBSITE**”, is a Bonafide work of **SUMIT DASHRATH GUPTA** bearing Seat. No: **556** submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY** from **Ramanand Arya D.A.V. College**, Autonomous, University of Mumbai for the academic year **2023-24**.

Internal Guide

Co-ordinator

External Examiner

Date:

College Seal

ABSTRACT

The “ **QR Code Generator** ” project aims to develop a versatile and user-friendly tool that enables the creation of Quick Response (QR) codes for various purposes.

QR codes have become an integral part of modern communication, allowing quick and efficient sharing of information through Smartphone cameras. This project will provide an intuitive interface where users can input different types of data, such as URLs, text, contact information, and generate corresponding QR codes.

The Website will focus on generating QR codes with high accuracy and customization, including options for adjusting colors, sizes, and error correction levels. By offering both web-based and mobile application, this project aims to cater to a wide range of users, from individuals to businesses, providing them with a seamless solution for generating QR codes tailored to their specific needs.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who helped me complete this project successfully and I came to know about so many new things.

First and foremost, I would like to thank to **CA Dr. Ravindra Naik, I/C Principal of Ramanand Arya DAV college**, who gave me golden opportunity to do this wonderful project. I would also like to thank **Mrs. Chandrakala Shrivastava (Co-ordinator of Self- Finance Program)** for her constant support, valuable guidance and encouragement throughout the process.

I would like to express my special thanks and gratitude to the project guide **Assistant Professor Amruta Joshi** for guiding me to do the project work on time and giving support and guidance, which made me complete my project duly. I am extremely thankful to her for providing such nice support and guidance.

I am thankful for and fortunate enough to get constant encouragement, support, and guidance from the teachers of Information Technology who helped me in successfully completing my project work.

Finally, I would like to thank my family who motivated me and boosted my morale.

DECLARATION

I hereby declare that the project entitled, “ **QR CODE GENERATOR** ” done at RAMANAND ARYA DAV COLLEGE, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

SUMIT DASHRATH GUPTA

Table of Contents

Sr. No.	Name	Page No.
	ABSTRACT	
	ACKNOWLEDGEMENT	
	DECLARATION	
	Chapter 1: Introduction	1-3
1.1	Background	1
1.2	Objectives	2
1.3	Purpose, Scope, Applicability	2
1.3.1	Purpose	2
1.3.2	Scope	2
1.3.3	Applications	2
1.4	Organization Report	3
	Chapter 2: Survey of Technologies	4-5
	Chapter 3: Requirements and Analysis	6-14
3.1	Problem Definition	6
3.2	Requirement Specification	6
3.2.1	Functional Requirements	6
3.2.2	Non-Functional Requirements	7
3.3	Planning and Scheduling	8
3.4	Software and Hardware Requirement	11
3.5	Modules Description	11
3.6	Future Development	12
3.7	Technology & Tools	13
	Chapter 4: System Design	15-30
4.1	UML Diagrams	15
4.1.1	Use Case Diagram	15
4.1.2	CLASS Diagram	17
4.1.3	ACTIVITY Diagram	19
4.1.4	COMPONENT Diagram	20
4.1.5	DATA FLOW Diagram	22
4.2	Data Design	24

4.2.2	Data Integrity and Constraints	25
4.3	Algorithm Design	26
4.4	Test Case Design	28
	Chapter 5: Implementation and Testing	31-35
5.1	Implementation and Approaches	31
5.2	Coding details and efficiency	31
5.2.1	Code Efficiency	32
5.3	Testing Approach	32
5.3.1	Manual Testing	33
5.3.2	Functional Testing	34
	Chapter 6: Result and Discussion	36-42
6.1	Test Reports	36
6.2	User Documentation	37
	Chapter 7: Conclusion	43-52
7.1	Conclusion	43
7.2	Future Scope of the project	44
	References	46

Chapter 1 : Introduction

Welcome to our QR Code Generator project, where we combine the power of HTML, CSS, and JavaScript React to create a user-friendly and efficient tool for generating QR codes. QR codes have become an essential part of modern communication, allowing users to quickly access information by scanning a simple code. In this project, we leverage the capabilities of HTML5 and CSS3 to design an intuitive user interface that enhances the user experience.

But that's not all – we take it a step further by implementing a secure login system using the robust technologies of Node.js, Express.js, and MongoDB. This login system ensures that only authorized users have access to the QR code generation functionality. Node.js provides the backend environment, Express.js streamlines server-side operations, and MongoDB handles the database management, all working together to create a seamless and secure user authentication process.

1.1 Background

"The QR Code Generator project is a versatile and user-friendly tool designed to simplify the process of creating QR codes. To enhance security and personalize the experience, the system features a robust login and signup mechanism. Users can easily register for an account or log in to access their QR code generation capabilities. Once logged in, the web-based interface provides a seamless experience for generating QR codes. Whether it's for business cards, product labels, or event promotions, users can effortlessly generate custom QR codes tailored to their specific needs. This project not only streamlines QR code creation but also ensures a secure and convenient user experience through its authentication system, making it a valuable asset for a wide range of applications.

1.2 Objective

- Efficient Information Sharing.
- User-Friendly Interface.
- Customization Options.
- Secure User Authentication.
- Scalability and Performance

1.3 Purpose, Scope, and Applicability

1.3.1 Purpose:

The purpose of this QR Code Generator project is to provide a user-friendly and secure system for generating QR codes after a user logs in or signs up. QR codes are versatile tools that can store various types of information, such as URLs, text, contact details, and more. By incorporating user authentication, this project ensures that QR codes are generated and accessed only by authorized users, enhancing data security and privacy.

1.3.2 Scope:

The project's scope includes the following key features and functionalities:

1. **User Authentication:** Users must log in or sign up before accessing the QR code generation functionality. This involves username/password authentication or alternative methods like social media login.
2. **QR Code Generation:** Once authenticated, users can generate QR codes with various content types, such as URLs, plain text, contact information, or any other relevant data.
3. **Customization:** Users may have options to customize the appearance of QR codes, such as colors and sizes, to suit their needs.

1.3.3 Applicability:

This QR Code Generator project is applicable in a wide range of scenarios, including but not limited to:

1. **Business Use:** Companies can use it to create QR codes for product information, marketing campaigns, and contactless payment options.

2. **Personal Use:** Individuals can generate QR codes for sharing contact information, Wi Fi access details, or event invitations.
3. **Education:** Teachers and educators can use it to create QR codes for links to educational resources and interactive learning materials.
4. **Event Management:** Event organizers can generate QR codes for event tickets ,attendee registration, and check-in processes.
5. **Data Sharing:** Users can use QR codes to securely share sensitive information ,such as medical records or private documents, with trusted parties.

1.4 Organization Report

Project Objectives:

- 1 .Develop a web-based application for generating QR codes.
- 2 .Ensure compatibility with various data types, including text, URLs, phone numbers, and contact information.
- 3 .Implement customization options for QR code design, such as color, size, and logo integration.
4. Ensure ease of use and intuitive interface for users of all technical levels.
- 5 ,Provide robust error handling and validation to ensure accurate QR code generation.
- 6 .Optimize the application for performance and scalability to accommodate high volumes of QR code generation .

Project Status:

The QR Code Generator project is currently in the development phase, with the team actively working on building and testing the web application. Progress is on track, and the project is expected to meet its scheduled milestones for deployment.

Chapter 2: Survey of Technology

Frontend Technologies:

1. **HTML:** Hypertext Markup Language is used for structuring the content of your web pages.
2. **CSS:** Cascading Style Sheets are used for designing and styling the appearance of your webpages.
3. **JavaScript:** A versatile scripting language used for interactivity and dynamic behavior on the client-side.

Framework:

React:

Description: React is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and manage the state of an application efficiently.

Key Features:

- **Declarative Syntax:** Developers describe the desired UI, and React takes care of updating the DOM to match that state.
- **Component-Based:** UIs are broken down into components, making code modular, reusable, and easier to maintain.
- **Virtual DOM:** React uses a virtual representation of the DOM to optimize rendering and improve performance.

Vite:

Description: Vite is a fast, opinionated web development build tool that aims to provide a faster and more streamlined development experience compared to traditional bundlers.

Key Features:

- **Lightning Fast:** Vite leverages native ES module imports to achieve fast development server start times and efficient hot module replacement (HMR).
- **No Bundling in Development:** Unlike traditional bundlers, Vite does not bundle your code during development, making the development server faster.

- **Pluggable:** Vite is designed to be extensible and supports various plugins for tasks like TypeScript compilation, CSS preprocessing, and more.
- **Vue and React Support:** While initially designed for Vue.js, Vite also supports React and other frameworks, making it versatile.

Using React with Vite:

- To use React with Vite, you typically initialize a new project with Vite and choose the React template. This sets up a project with the necessary configurations for using React and Vite together seamlessly.
- Developers can then create React components and build the application using Vite's fast development server and optimized build process.

Backend Technologies:

1. **Node.js:** A runtime environment that allows you to run JavaScript on the server-side, making it possible to build web applications with a unified language (JavaScript).
2. **Express.js:** A web application framework for Node.js that simplifies the process of building robust and scalable web applications by providing a set of powerful features and tools.
3. **MongoDB:** A NoSQL database system that is used for storing and managing your application's data in a flexible and schema-less manner.

Chapter 3: Requirement and Analysis

3.1 Project Definition

The QR Code Generator with Login and Signup System is a web-based application designed to allow users to create QR codes for various purposes. This project combines the functionality of generating QR codes with a user authentication system that includes login and signup capabilities. Users can access the QR code generation features after creating an account or logging in.

3.2 Requirements Specification (SRS)

Requirement specification for an online library management system outlines the functional and non-functional requirements that the system should meet. Here's a possible requirement specification for the project: -

3.2.1 Functional Requirements:

1) QR Code Generation:

- The system should be able to generate QR codes based on user input or data provided.
- Support for various types of QR codes (URL, text, contact information, etc.).

2) User Input:

- Provide a user-friendly interface for users to input data for QR code generation.
- Allow users to customize QR code features such as size, color, and error correction level.

3) Data Validation:

- Validate input data to ensure it adheres to the specifications of the selected QR code type.

4) Code Export:

- Provide an option for users to download the QR code image directly.

3.2.2 Non-Functional Requirements:

1) Performance:

- The system should generate QR codes quickly, even when dealing with large datasets.
- The time to scan and extract information from QR codes should be minimal.

2) Scalability:

- The system should be scalable to handle a growing number of users and QR code generation requests.

3) Reliability:

- The QR codes generated should be reliable and accurately encode the provided information.

4) Security:

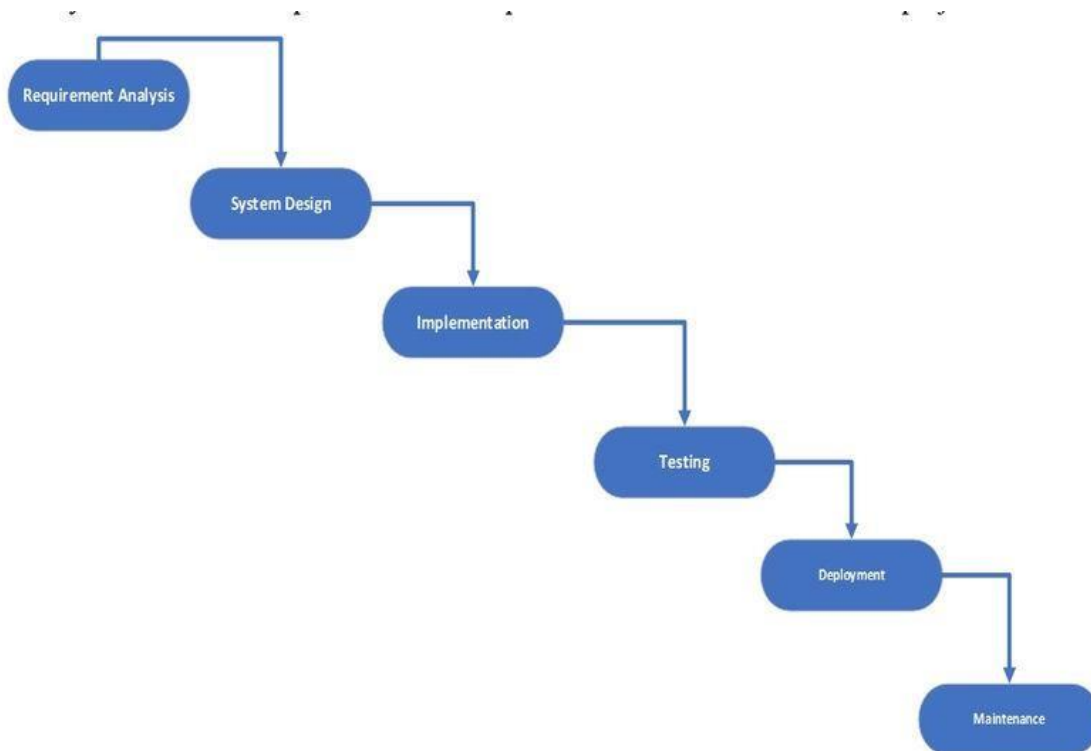
- Implement security measures to protect sensitive user data used in QR code generation.
- Ensure that QR codes cannot be easily tampered with or forged.

3.3 Planning and Scheduling

Planning and scheduling can be termed as most complicated part of software development. Success and failure of a project largely depends on how it has been planned, while its timely completion depends on its scheduling. Planning, for our purposes, can be thought of determining all the small tasks that must be carried out in order to accomplish the goal. Planning also takes into account, rules, and known as constraints, which, control when certain task can or cannot happen? Scheduling can be thought of as determining whether adequate resources are available to carry out the plan. Improper planning can lead to unusual late release of software which is not desirable.

❖ Waterfall Model

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.



- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

❖ Gantt Chart

A Gantt chart is a popular project management tool that provides a visual representation of a project schedule. It was developed by Henry L. Gantt in the early 20th century and is widely used to plan, schedule, and track the progress of tasks and activities within a project. Here's an explanation of the key components and how Gantt charts work:

Task List:

The first step in creating a Gantt chart is to list all the tasks or activities that need to be completed for the project. These tasks should be broken down into manageable and specific components.

Time Scale:

A Gantt chart typically has a horizontal time scale represented along the top or bottom of the chart. This scale could be days, weeks, months, or any other relevant time unit depending on the project's duration and complexity.

Task Duration:

Each task is represented as a horizontal bar on the chart, positioned along the time scale. The length of the bar corresponds to the duration of the task. Longer bars indicate longer task durations.

Dependencies:

Gantt charts often include dependencies, which indicate the relationships between tasks. Dependencies define the order in which tasks need to be completed. For example, Task B may depend on the completion of Task A before it can start.

Milestones:

Important project milestones, which are significant points of progress, are often marked on the Gantt chart. Milestones help project managers and team members track key events and achievements.

Resource Allocation:

Some Gantt charts include information about resource allocation, showing which resources (such as team members or equipment) are assigned to each task.

Progress Tracking:

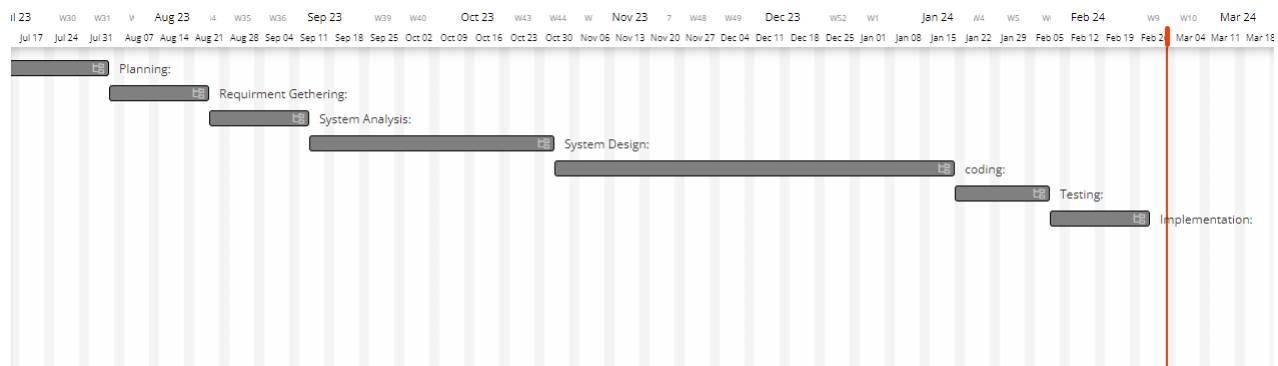
As the project progresses, the Gantt chart is updated to reflect the actual progress made on each task. This allows for a quick visual assessment of whether the project is on schedule or if adjustments need to be made.

Critical Path:

The critical path is the sequence of tasks that determines the overall duration of the project. It represents the longest path through the project and identifies tasks that, if delayed, could impact the project's completion date.

Color Coding:

Gantt charts often use color coding to distinguish between different types of tasks, resources, or phases of the project.



3.4 Software & Hardware

Requirements Software

Requirement:

- **Frontend:** Html , CSS , JavaScript , React with Vite .
- **Backend:** Node.js , Express.js , MongoDB
- **Platform:** Windows10

Hardware Requirement:

- **Processor:** Intel(R) Core(TM) i3-10105 CPU @ 3.70GHz
- **RAM:** 8.00 GB(Minimum)
- **Physical Memory (HDD):**500GB
- **Input Peripherals:** Keyboard & Mouse

3.5 Modules Description

3.5.1 User Authentication and Management:

- **User Registration:** Allow users to create accounts using their email addresses and secure passwords.
- **User Login:** Provide a secure login mechanism for authenticated access to the QR code generation features.

3.5.2 QR Code Generation:

- **Data Input:** Provide an interface for users to input the data they want to encode into a QR code, such as URLs, text, contact information, etc.
- **QR Code Customization:** Allow users to customize the appearance of the generated QR code, including options like colors, logos, and size adjustments.

- **QR Code Download:** Generator QR Code than Download In Your Folder.
- **QR Code Generation:** Implement the core functionality to convert user input in to a QR code image using JavaScript.

3.6 Future Development

1. **Enhanced QR Code Types:** Expand the range of QR code types that the generator can handle. Currently supporting URLs, text, and contact information, you could extend it to support more data types like Wi-Fi credentials, event details, and location coordinates.
2. **Batch QR Code Generation:** Implement a feature to generate multiple QR codes at once. This could be useful for businesses or events that require generating a set of codes with similar information, such as discount codes or attendee badges.
3. **QR Code Analytics:** Integrate tracking and analytics to monitor QR code scans. This would provide valuable insights into how often codes are being used, where they are being scanned, and other user behavior patterns.
4. **Mobile App Integration:** Create a mobile app version of the QR Code Generator, allowing users to generate QR codes on the go. This could include features like using the device's camera to directly scan content for the QR code.
5. **Social Media Sharing:** Integrate sharing options that allow users to directly share generated QR codes on popular social media platforms. This could facilitate the dissemination of information to a wider audience.

3.7 Technology & Tools

Technology:

HTML (Hypertext Markup Language): HTML is the standard markup language used to create the structure and content of web pages. It defines the elements and layout of a web page.

CSS (Cascading Style Sheets): CSS is used to control the presentation and styling of HTML elements. It allows you to apply colors, fonts, spacing, and other visual aspects to create a visually appealing user interface.

JavaScript: JavaScript is a versatile programming language that enables interactivity and dynamic behavior in web pages. It can be used to manipulate the DOM (Document Object Model), handle user input, perform animations, and communicate with server

React with Vite :

- To set up a React project with Vite, you typically use a template or initialize a project using a command like create – Vite .
- Vite's development server provides fast HMR, making the development process smoother.
- Vite uses a modular and optimized build system that generates highly optimized production-ready code.
- You can leverage Vite plugins to extend its functionality and customize the build process further

Node.js: Node.js is a runtime environment that allows you to run JavaScript on the server-side. It enables building scalable and high- performance applications using JavaScript, which was traditionally limited to the browser.

Express.js: Express.js is a web application framework for Node.js. It simplifies the process of building web applications by providing a set of tools and utilities for routing, middle ware ,and handling HTTP requests and responses.

MongoDB: MongoDB is a popular NoSQL database that stores data in a flexible, JSON- like format called BSON (Binary JSON). It's designed to handle large amounts of unstructured or semi-structured data and is commonly used in modern web applications.

Tools:

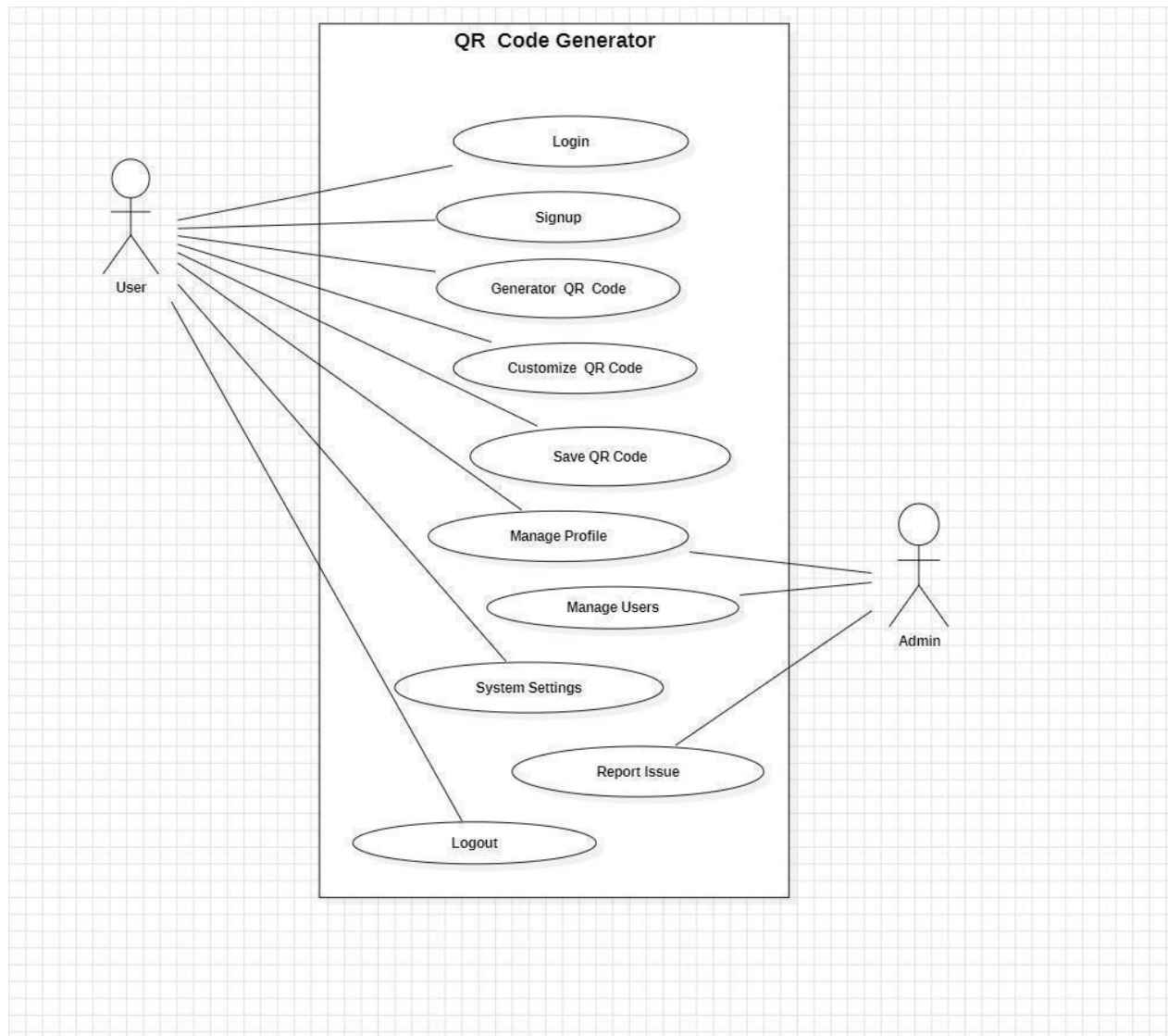
Visual Studio Code: Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and NodeJS and has a rich ecosystem of extension for other languages and runtimes (such as C++, C, Java, Python, PHP, .NET, etc.).

Overall: Visual Studio Code is a highly versatile and widely used code editor suitable for various programming tasks, from writing simple scripts to building complex applications. Its rich feature set, extensions, and customization options make it a popular choice among developers.

Chapter 4: System Design

4.1 UML Diagram

4.1.1 Use Case Diagram :



User: This is the primary actor who interacts with the system.

Generate QR Code: This use case represents the core functionality of the system, where the user can generate QR codes.

Customize QR Code: This use case allows the user to customize the color and size of the generated QR code.

Login: This use case represents the user's ability to log in to the system.

Sign Up: This use case allows new users to create an account.

Save QR Code: Users can save the generated QR code to their account.

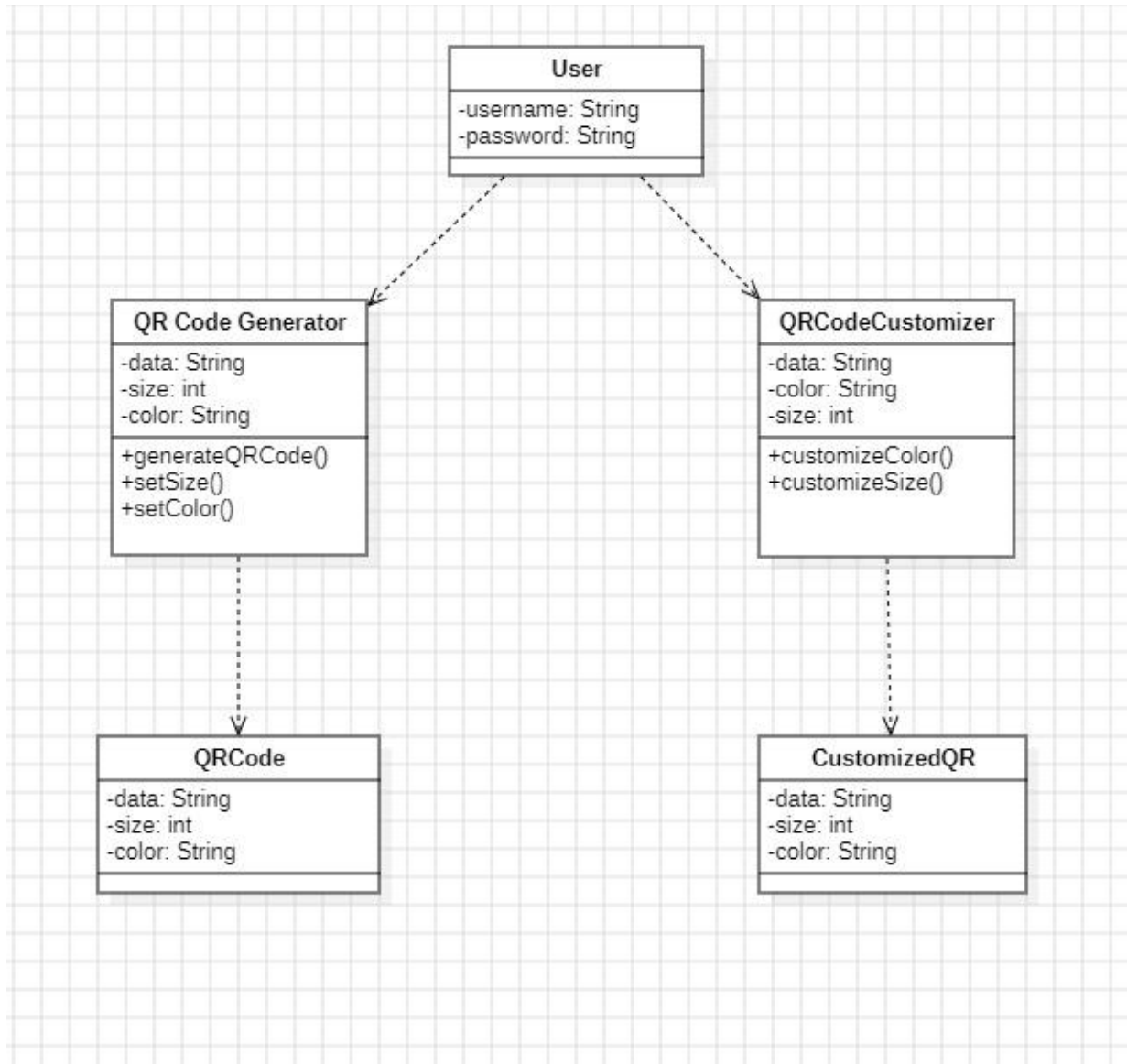
Retrieve QR Code: This use case allows users to retrieve previously saved QR codes.

Logout: Users can log out of the system when they are done.

Arrows connecting the actors to the use cases indicate who can perform or interact with each use case. For example, "User" connects to "Generate QR Code," "Customize QR Code," "Login," "Sign Up", "Save QR Code," "Retrieve QR Code," and "Logout."

This use case diagram outlines the main interactions and functionalities of a QR code generator system with customization options and a login/signing system.

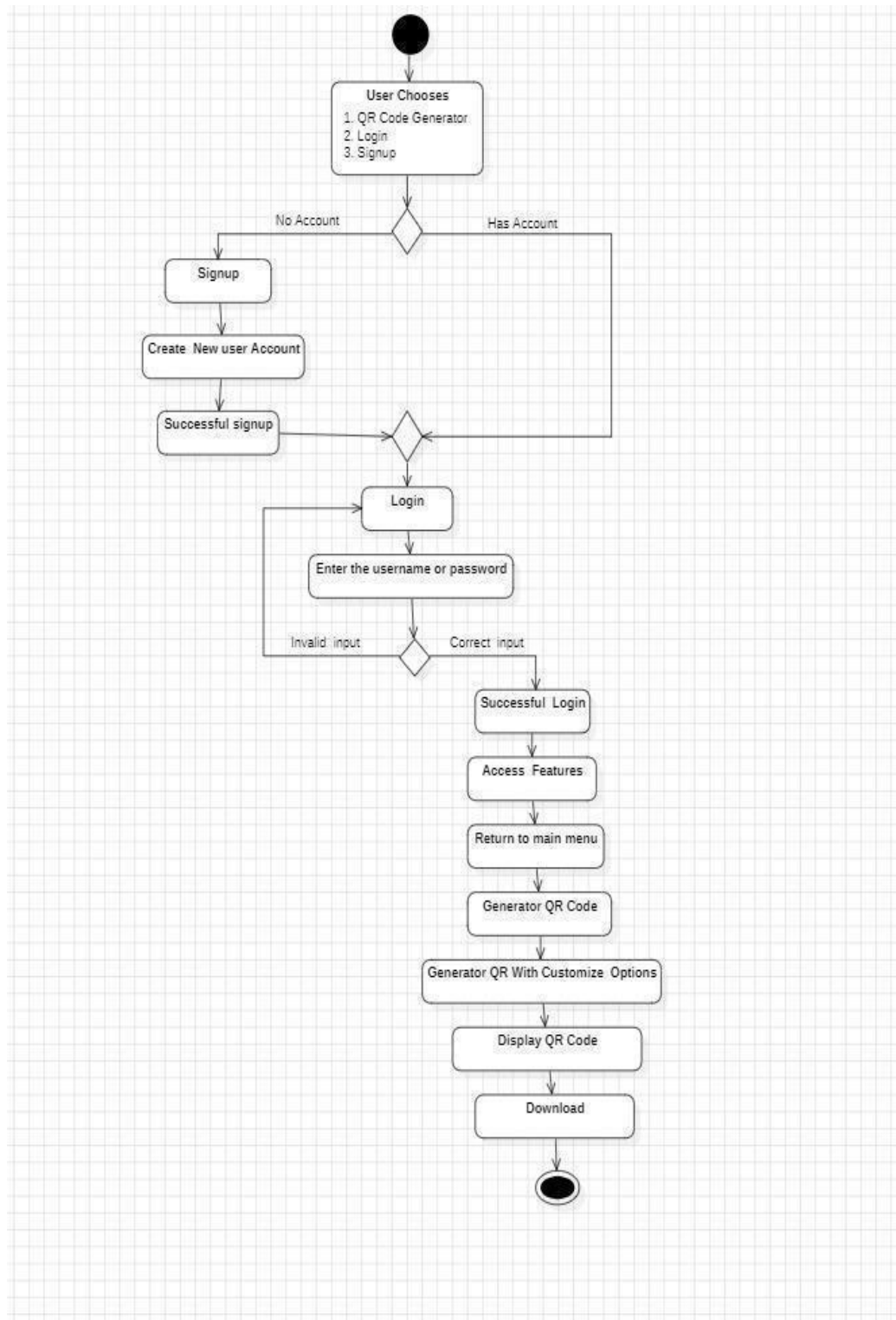
4.1.2 Class Diagram :



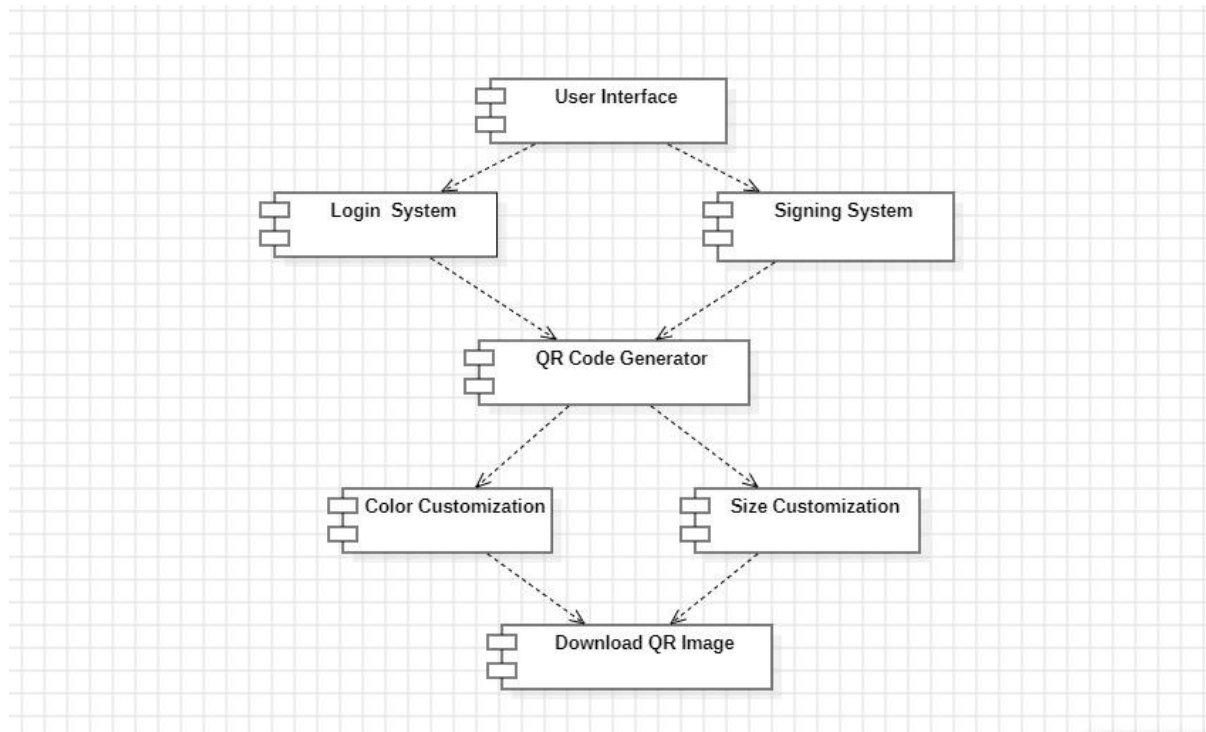
1. **QR Code Generator:** This class is responsible for generating standard QR codes .It has attributes for data, size, and color. Methods include generate QR Code(), set Size(), and set Color().

2. **QR Code Customizer:** This class is responsible for customizing QR codes by changing their color and size. It has attributes for data, color, and size. Methods include ``customize Color()`` and ``customize Size()``.
3. **User:** Represents a user in the system, with attributes for username and password.
4. **QR Code and Customized QR:** These classes represent the generated QR codes. ``Customized QR`` inherits from ``QR Code`` and adds customization attributes. These classes contain attributes for data, size, and color. Arrows indicate associations between classes, such as ``QR Code Generator`` creating instances of ``QR Code``. Additionally, there would likely be methods to handle user authentication and storage of customized QR codes within the ``User`` and ``User Manager`` classes, but those are not shown in the diagram for simplicity.

4.1.3 Activity Diagram :



4.1.4 Component Diagram :



1. User Interface (UI):

-This component represents the user interface where users interact with the system. Users can input data and customize QR codes here.

2. QR Code Generator:

-This core component generates QR codes based on user inputs and customization preferences.

3. Color Customization:

-This component allows users to customize the color of the generated QR code, which is part of the QR code's design.

4. Size Customization:

-This component allows users to customize the size or dimensions of the generated QR code.

5. Login System:

-This component manages user authentication and authorization. Users may need to log in before generating QR codes or accessing certain features.

6. Signing System:

- This component manages digital signatures for QR codes or user actions. It ensures the integrity and authenticity of QR codes or user-generated content.

The arrows in the diagram represent the flow of data and interactions between the components. For example:

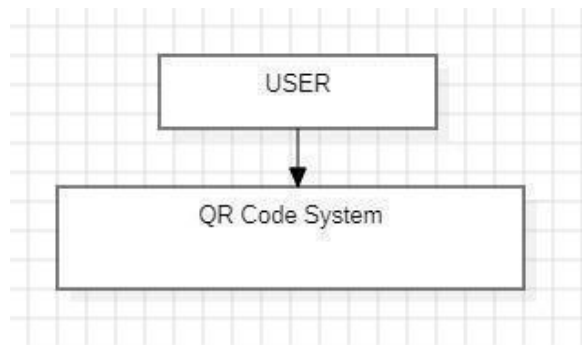
- User inputs data and customization preferences through the UI.
- The UI sends this data to the QR Code Generator.
- The QR Code Generator may interact with Color Customization and Size Customization components to apply user preferences.
- If the user has to log in, the Login System verifies their identity.
- The Signing System may be used to add digital signatures to generated QR codes or authenticate user actions.

This is a simplified representation, and in a real-world application, these components may interact with databases, external services, and other components. The diagram can be further refined based on the specific requirements and architecture of your QR code generator and login system

4.1.5 Data Flow Diagram :

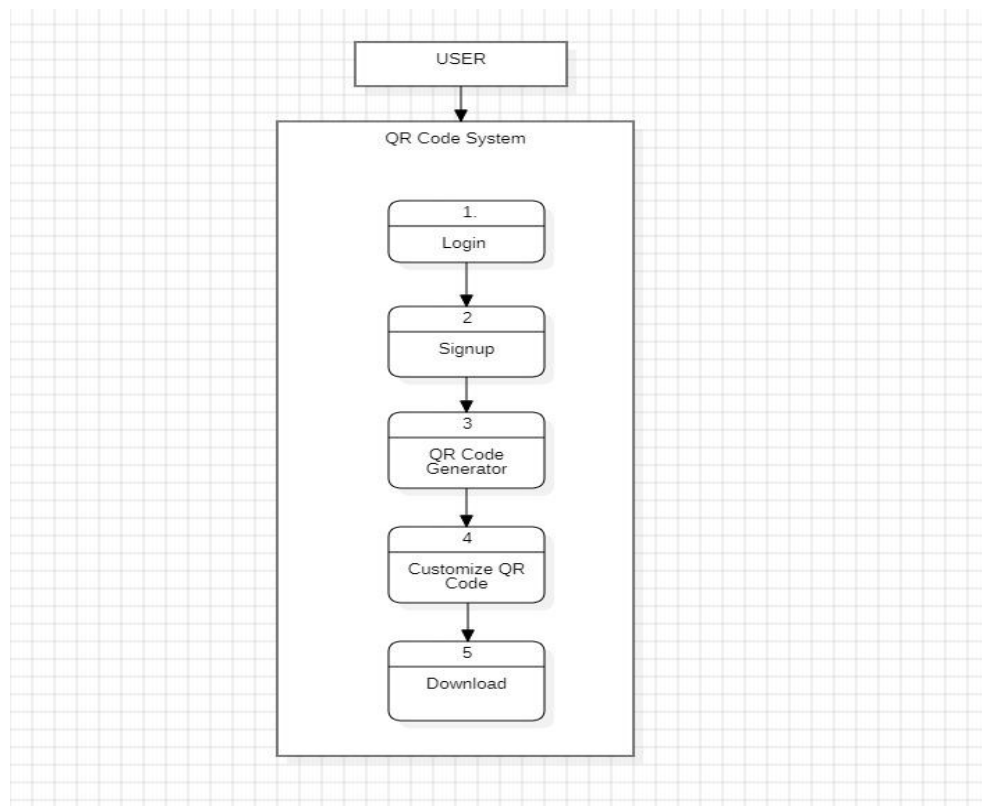
Level 0 DFD (Context Diagram)

In a context diagram, you show the system as a single process and the external entities interacting with it.



Level 1 DFD

Now, let's break down the QR code system into more detailed processes and dataflows.



1. **Login (Process 1):** Users can log in to the system.
 - Data Flow: User credentials (username and password) → Authentication process
2. **Signup (Process 2):** New users can create accounts.
 - Data Flow: User registration data → User account creation process
3. **QR Code Generator (Process 3):** This process generates QR codes.
 - Data Flow: User data (if needed) → QR code generation process
4. **Customize QR Code (Process 4):** Users can customize QR codes by specifying color and size.
 - Data Flow: Customization data (color and size preferences) → QR code customization process

These processes interact with external entities (Users) and may involve data storage, retrieval, and processing. You can further expand the details of each process, including the specific data elements and interactions within each process. Additionally, you would need to define the data stores (databases or file systems) where user data, QR codes, and customization preferences are stored and retrieved.

This DFD provides an overview of the key components and data flows in your QR code generator system, including user management and customization options.

4.2 DATA DESIGN:

Data design is process of designing a database. the main output of a data design is a detailed logical data model of database. A logical data model is one of the main outputs of data design. The data model is usually represented as an entity relationship diagram or ER diagram. While a person can do both data design and database analysis, these are two different tasks. Database analysis takes that model and applies it to one or more database engines.

4.2.1 SCHEMA DESIGN:

Database descriptions are called database schemas that are specified during database design and are not expected to change frequently. Most Data models have specific conventions for presenting schemes such as diagrams. The schematic presented is called the schematic diagram. Database schema design organizes the data into separate entities, determines how to create relationships between organized entities, and how to apply the constraints on the data.

The term “schema” refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). In MongoDB we create schema in key value pairs.

Registration:

Email	Password
-------	----------

Login:

Email	Password
-------	----------

Message:

First_name	Last_name	Email	Phone	message
------------	-----------	-------	-------	---------

1.1.1 DATA INTEGRITY AND CONSTRAINTS :

Data integrity is the maintenance and the assurance of the accuracy and consistency of data over its entire lifecycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data.

Registration:

Keys	Data Types	Constraints
Email	String	Required
Password	String	Required

Login:

Keys	Data Types	Constraints
Email	String	Required
Password	String	Required

Message:

Keys	Data Types	Constraints
First_name	String	Required
Last_name	String	Required
Email	String	Required
Phone	Number	Required
Message	String	Required

4.3 Algorithm for QR Code Generation:

Input:

- User provides data to be encoded in the QR code.
- User specifies QR code type (URL, text, contact information, etc.).
- User may customize optional parameters such as size, color, and error correction level.

Data Validation:

- Validate input data to ensure it adheres to the specifications of the selected QR code type.
- Handle errors gracefully, providing feedback to the user.

Encoding:

- Use the appropriate encoding method based on the QR code type selected.
- Apply error correction encoding as per the chosen error correction level.

Module Placement:

- Arrange encoded data modules within the QR code grid.
- Position format information and alignment patterns.

Version and Mask Selection:

- Determine the appropriate QR code version based on the amount of data to be encoded.
- Choose the best mask pattern to improve QR code readability.

QR Code Generation:

- Combine data modules, format information, and alignment patterns to generate the final QR code matrix.

Rendering:

- Convert the QR code matrix into an image format (e.g., PNG or JPEG).
- Apply user-specified customizations such as size and color.

Output:

- Display the generated QR code to the user.
- Provide options for the user to download the QR code image.

Pseudocode:

function generate QR Code(data, type, options):

 validate Data(data, type)

 encoded Data = encode Data(data,
 type) arrange Modules(encoded
 Data) select Version And
 Mask(encoded Data) generate QR
 Code Matrix(encoded Data) render
 QR Code Matrix(options)

 Display QR Code()

 offer Download Option()

USER INTERFACE

Design The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse, and the appearance of a desktop. It is also the way through which a user interacts

with an application or a website. UI stands for User Interface. UI is the part of the web application which a user interacts with. In simple terms, it is everything you see and touch, such as buttons, colors, fonts, navigation, etc.

4.4 TEST CASE

The Test Case is the set of conditions or variables under which a tester will determine whether the system under test satisfies requirements or works correctly. The process of developing test cases can also help to find problems in the requirements or design of the application. Test case design refers to how you set up your test cases. It is important that your tests are designed well, or you could fail to identify bugs and defects in your software during testing. There are many different test case design techniques used to test the functionality and various features of your software.

- **Login Test Cases:**

Test case No.	Test Case Name	Test Case Description	Test Case	
			Steps	Expected
TC1	Validate email and password	To verify email	The email field is left empty or the email is incorrect.	An error messages “Enter email”
		To verify password	The password field is left empty or the password is incorrect.	An error messages “Enter password”

- **Registration Test Cases:**

Test case No.	Test Case Name	Test Case Description	Test Case	
			Steps	Expected
TC2	Validate email and password	To verify email	The email field is left empty or the email is incorrect.	An error messages “Enter email”
		To verify password	The password field is left empty or the password is incorrect.	An error messages “Enter password”

- **Message Test Case:**

Test Case No.	Test Case Name	Test Case Description	Test Case	
			Steps	Expected
TC3	Validate First_name	To verify First_name	The first_name field is empty	An error messages “Enter first_name”
TC4	Validate last_name	To verify last_name	The last_name field is empty	An error messages “Enter last_name”
TC5	Validate email	To verify email	The email field is empty	An error message “Enter mail”
TC6	Validate Phone_no	To verify Phone_no	The Phone_no field is empty	An error messages “Enter Phone_no”
TC7	Validate Message	To verify Message	The Message field is empty	An error messages “Enter Message”

- **QR Code Generator Test Case:**

Test Case No.	Test Case Name	Test Case Description	Test Case	
			Steps	Expected
TC8	Validate Text or URL	To verify Text or URL	The Generator QR code on the screen	An error messages “Enter URL Name”

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation And Approaches

Project implementation (or project execution) is the phase where visions and plans become reality. This is the logical conclusion, after evaluating, deciding, envisioning, planning, applying for funds, and finding the financial resources of a project. Technical implementation is one part of executing a project.

An implementation plan for a project refers to a detailed description of actions that demonstrate how to implement an activity within the project in the context of achieving project objectives, addressing requirements, and meeting expectations.

5.2 Coding Efficiency And Details

Modular Code:

Break your code into modular functions to improve readability and maintainability.

Error Handling:

Implement error handling to check for empty input or other potential issues.

Comments:

Add comments to explain complex logic or functions to make the code More understandable for other developers.

User Interface:

Design a user-friendly interface with clear instructions and feedback.

Responsive Design:

Ensure that your web application is responsive and works well on various devices.

Testing:

Perform thorough testing to identify and fix any bugs or issues.

Security:

If you plan to deploy the project, consider security aspects, especially if dealing with user-generated content.

Performance:

Optimize your code for performance, especially if dealing with large datasets.

5.3 Testing

Approach Unit

Testing:

Test individual functions in your JavaScript code to ensure they work correctly.

Example: Write tests to check if the generate QR Code() function produces the correct QR code for different inputs.

Integration Testing:

Test the interaction between different components of your application. **Example:** Ensure that the input from the text field is correctly processed and used to generate the QR code.

Functional Testing:

Test the overall functionality of the QR code generator.

Example: Test the entire process from entering text to generating and displaying the QR code.

User Interface Testing:

Verify that the user interface is responsive and works well on different devices and browsers.

Example: Check if the input field, button, and QR code display are correctly styled and positioned.

Cross-browser Testing:

Test the application on different browsers to ensure compatibility. **Example:** Verify that the QR code generator works on popular browsers like Chrome, Firefox, and Safari.

Performance Testing:

Test the application's performance, especially when dealing with large datasets. **Example:** Measure the time it takes to generate and display a QR code for different input sizes.

Test Step :

5.3.1 Login Test Cases:

Testcase ID	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
TC1	Validate email and password	To verify email	Username and Password	Display success “Enter email”	The email field is left empty or the mail is incorrect.	Pass
		To verify password		Display success “Enter password”	The password field is left empty or the password is incorrect.	

5.3.2 Registration Test Cases:

Test case ID	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
TC2	Validate email and password	To verify email	Username and Password	Display success “Enter email”	The email field is left empty or the mail is incorrect.	Pass
		To verify password		Display success “Enter password”	The password field is left empty or the password is incorrect.	

5.3.3 Message Test Case:

Testcase ID	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
TC3	Validate First_name	To verify First_name	Username First name	Display success first name	The first_name field is empty	Pass
TC4	Validate last_name	To verify last_name	Username Last name	Display success name last_name”	The last_name field is empty	Pass

TC5	Validate email	To verify email	Message email	Display success “Enter email”	The email field is empty	Pass
TC6	Validate Phone_no	To verify Phone_no	Message Phone no	Display success “Enter Phone_no”	The Phone_no field is empty	Pass
TC7	Validate Message	To verify Message	Message	Display success “Enter Message	The Message field is empty	Pass

5.3.4 QR Code Generator Test Case:

Test case ID	Test Scenario	Test Case	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
TC8	Validate Text or URL	To verify Text or URL	Text Decode	Display success “Enter URL Name”	The Generator QR code on the screen	Pass
TC9	Your QR Code Image Download	Download Image	Text to convert QR code generator	Display success Image	QR Code Image in form of png	Pass

CHAPTER 6: RESULT AND DISCUSSION

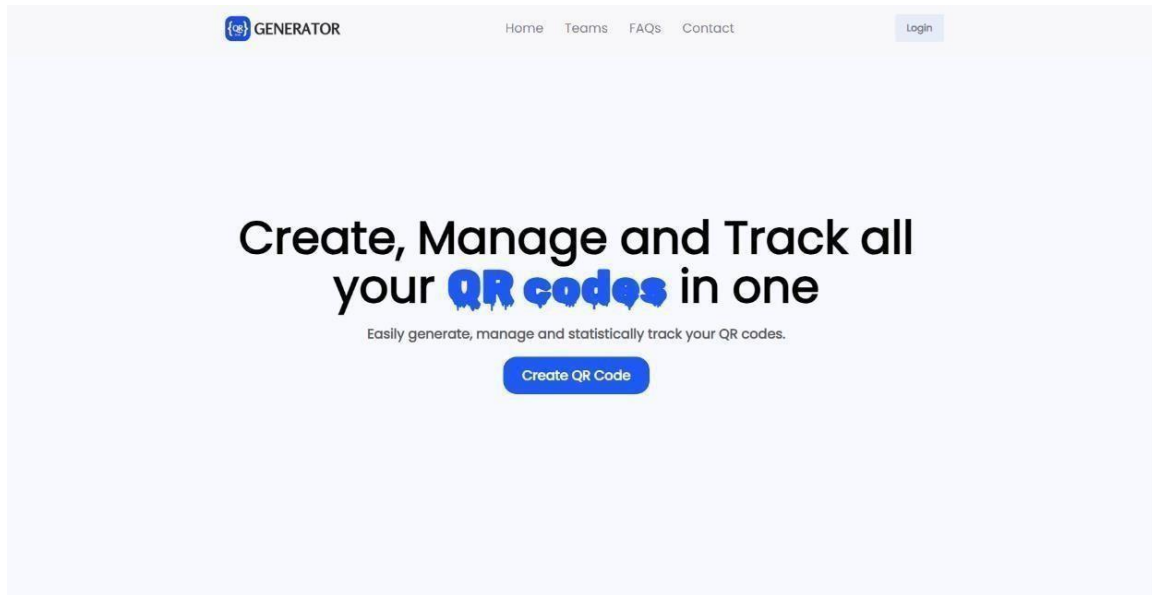
6.1 TEST Reports

Test Report is an important deliverable that is prepared at the end of a Testing project, or rather after Testing is completed. The prime objective of this document is to explain various details and activities about the Testing performed for the Project, to the respective take holders like Senior Management, Client, etc.

Test Report					
<i>Test CaseID</i>	<i>Description</i>	<i>Testing Approach</i>	<i>Total Test Cases Executed</i>	<i>% Test Case Passed</i>	<i>Priority</i>
T1	Check user login	User accounts Login	1	100%	High
T2	Check user account creation and registration	User accounts werecreated and registration	1	100%	High
T3	Check First_name	Validate First_name	1	100%	High
T4	Check last_name	Validate last_name	1	100%	High
T5	Check email	Validate email	1	100%	High
T6	Check Phone_no	Validate Phone_no	1	100%	High
T7	Check Message	Validate Message	1	100%	High
T8	Check correct URL or text or not	Validate Text or URL	1	100%	High
T9	Your QR Code Image Download	The Download Button	1	100%	High

6.2 USER Documentation

- **Home**



Headline:

- Create, Manage and Track all your QR codes in one .

Subheading:

- Easily generate, manage and statistically track your QR codes.

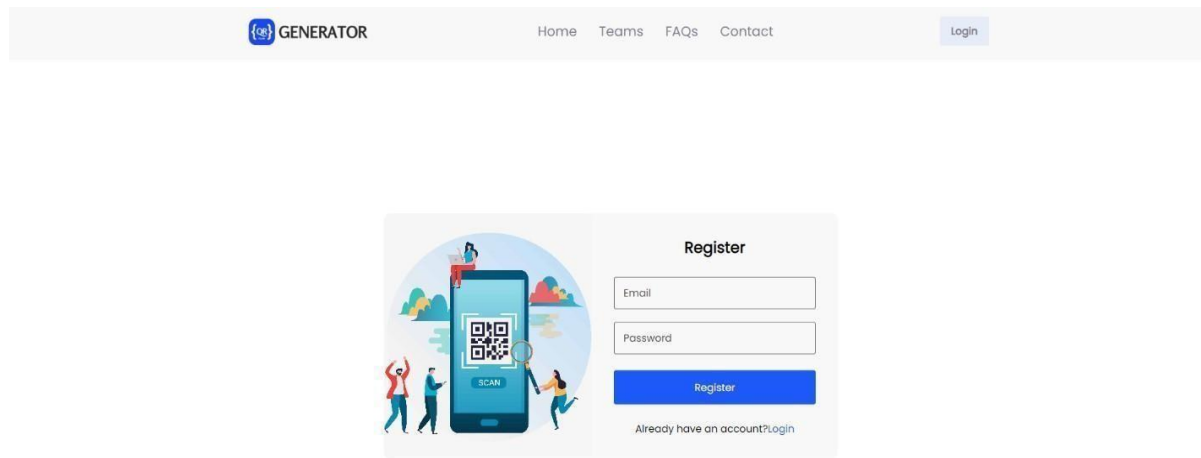
Body copy:

- QR codes are a powerful tool that can be used for a variety of purposes, such as tracking inventory, sharing contact information, and marketing products. With our QR code generator, you can easily create QR codes for any purpose.
- Our QR code generator is easy to use and allows you to create QR codes in a variety of formats. You can also track the performance of your QR codes, so you can see how many people are scanning them.
- Sign up for a free account today and start creating your QR codes!

Call to action:

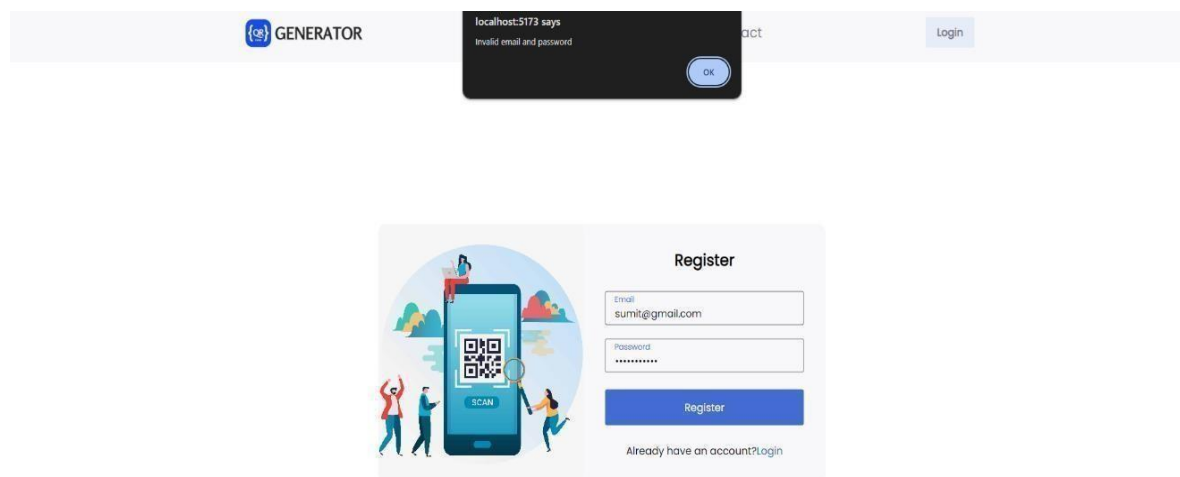
- Create QR Code

- **Registration**



The screenshot shows the top navigation bar of the GENERATOR website. It includes the logo, links for Home, Teams, FAQs, and Contact, and a Login button. Below the navigation bar is a registration form titled "Register". The form has two input fields: "Email" and "Password". Below these fields is a blue "Register" button. At the bottom of the form, there is a link that says "Already have an account? Login". To the left of the form is an illustration of a large smartphone displaying a QR code, with several people around it, some scanning the code.

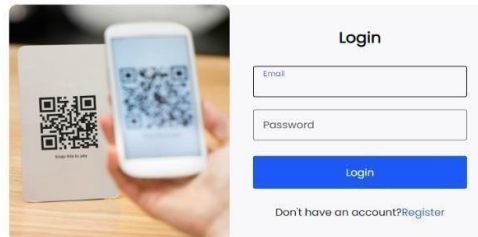
Registration is the very first process for creating an account which will be further used for the login process. It can be done by entering the correct credential.



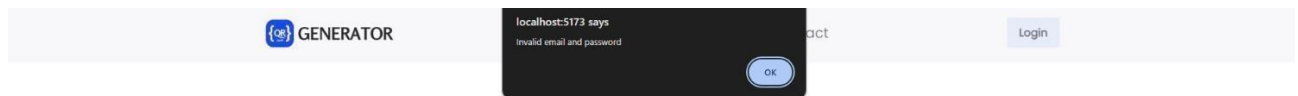
This screenshot shows the same registration page as above, but with an error message displayed. A dark modal box in the center of the page contains the text "localhost:5173 says" and "Invalid email and password". There is an "OK" button at the bottom right of the modal. The registration form below the modal shows the "Email" field filled with "sumit@gmail.com" and the "Password" field filled with "*****". The "Register" button is still visible.

It appears to be a login page for a user account on the website "localhost:5173". The page shows a message indicating an "Invalid email and password". There are also fields for entering an email address and password, as well as buttons to "Register" and "Login".

- Login



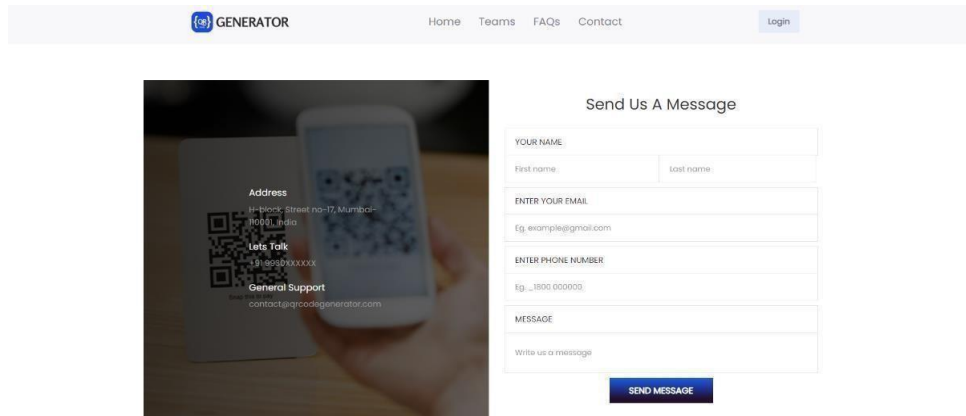
The Login is the second process after the registration. The login can be done by entering the correct credentials such as username and password which will further check the credentials from registration and if the credentials are correct then it will redirect to Dashboard.



The page shows an error message indicating "Invalid email and password". There are also fields for entering an email address and password, as well as buttons to "Register" and "Login".

I can't provide any information about the specific functionality of this webpage as it appears to be an internal tool and not a publicly available website.

- **Message**



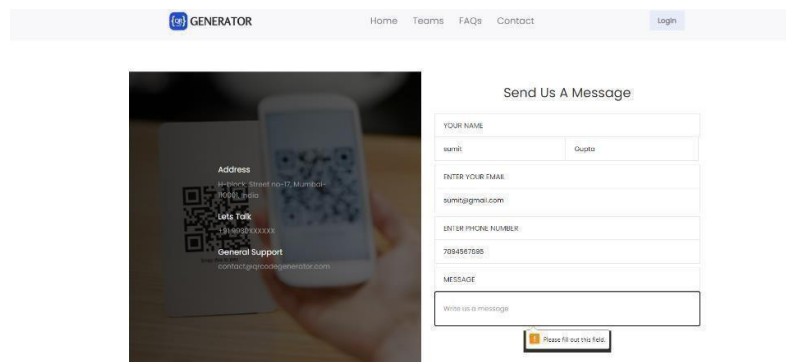
The screenshot shows the top of the QR Code Generator website. The header includes the logo, navigation links (Home, Teams, FAQs, Contact), and a Login button. Below the header, there is a QR code for general support with contact details. To the right, the 'Send Us A Message' form is displayed, featuring fields for name, email, phone number, and a message box, along with a 'SEND MESSAGE' button.

Header:

- The logo of QR Code Generator
- Menu bar with options like "Home", "Teams", "FAQs", "Contact", and "Login"

Body:

- A hero section with the text "GENERATOR"
- A search bar to enter a message to create a QR code
- A button to "Send Us A Message" and a form with fields to enter your name, address, email address, phone number, and message
- Information about General Support with contact details like phone number and email address



This screenshot is similar to the one above but includes a 'Please fill out this field.' error message below the message input field. The form fields are: YOUR NAME (first and last), ENTER YOUR EMAIL (example: sumit@gmail.com), ENTER PHONE NUMBER (708487088), and MESSAGE (placeholder: Write us a message). The 'SEND MESSAGE' button is at the bottom.

Body:

- A hero section with the text "Send Us A Message" and a form with fields to enter your name, address, email address, phone number, and message.
- Information about Contact Center support with a phone number

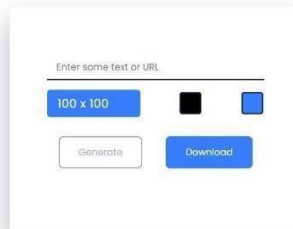
• QR Code Dashboard

Functionality:

The form allows users to generate QR codes by entering text or a URL in the designated field. Users can also choose the size of the QR code in pixels. Once they've entered their desired information, they can click the "Generate" button to create the QR code.

Additional details:

- The form also offers a "Download" button, which presumably allows users to download the generated QR code as an image file.
- There is a text box labeled "Enter some text or URI" where users can enter the data they want to encode in the QR code.
- Below the text box is a drop-down menu labeled "100 x 100", which likely allows users to select the size of the generated QR code in pixels.



A screenshot of a web form titled "QR Code Dashboard". At the top, there is a text input field with the placeholder text "Enter some text or URI". Below the input field, there is a dropdown menu currently showing "100 x 100". To the right of the dropdown menu, there are two small square color selection buttons, one black and one blue. Below these elements, there are two buttons: a light blue "Generate" button and a blue "Download" button.

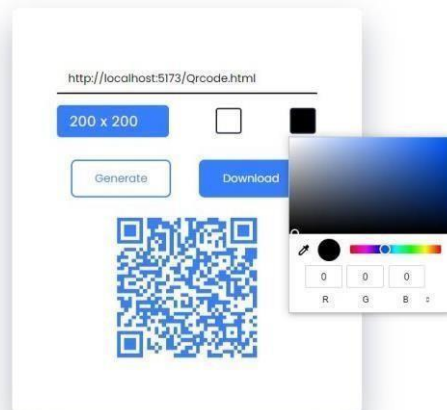
• QR code Generator

STEP 1 :

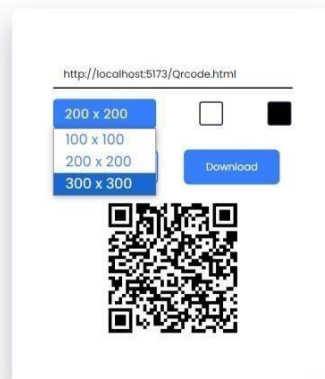


A screenshot of the "QR code Generator" form. At the top, there is a text input field containing the URL "http://localhost:5173/Qrcode.html". Below the input field, there is a dropdown menu currently showing "200 x 200". To the right of the dropdown menu, there are two small square color selection buttons, one black and one blue. Below these elements, there are two buttons: a light blue "Generate" button and a blue "Download" button. Below the buttons, a QR code is displayed, which encodes the URL entered in the input field.

STEP 2 :



STEP 3 :



CHAPTER 7: CONCLUSION

In the concluding chapter of the project, we summarize the key findings, achievements, and insights gained throughout the development of the Free QR Code Generator. This chapter serves as a wrap-up, providing closure to the project and highlighting its significance.

7.1 CONCLUSION

Objective Achievement:

Begin by revisiting the initial objectives set for the project. Evaluate how well the Free QR Code Generator met these goals and discuss any challenges faced along the way. Highlight specific features and functionalities that were successfully implemented.

Technical Insights:

Discuss the technical aspects of the QR code generator. This could include the programming languages, frameworks, and tools used in the development process. Provide insights into any innovative solutions or techniques applied to overcome technical challenges.

User Experience and Interface Design:

Evaluate the user experience (UX) of the Free QR Code Generator. Discuss the design choices made to enhance user-friendliness and accessibility. Showcase any user interface improvements made based on feedback or usability testing.

Testing and Quality Assurance:

Detail the testing processes employed during the development lifecycle. Discuss how the QR code generator was tested for functionality, performance, and security. Highlight any measures taken to ensure the quality and reliability of the tool.

User Feedback and Iterative Development:

Share feedback received from users during the beta testing phase or any user feedback mechanisms implemented. Discuss how this feedback influenced iterative development and improvements made to address user needs and concerns.

Future Enhancements and Development Roadmap:

Outline potential future enhancements and features that could be added to the Free QR Code Generator. Consider emerging technologies, user demands, and industry trends. Provide a roadmap for the project's future development, including any planned updates or expansions.

Open Source Collaboration and Community Engagement:

If applicable, discuss any plans for making the QR code generator an open-source project. Highlight the benefits of community collaboration and the potential for contributions from the developer community.

Conclusion and Acknowledgments:

Summarize the overall success of the Free QR Code Generator project. Acknowledge the efforts of the development team, express gratitude for any external support or collaborations, and recognize the contributions of key individuals.

Call to Action:

Encourage users to provide additional feedback, report any issues, or contribute to the project's ongoing development. Provide relevant contact information or links to community forums where users can engage with the project.

7.1 Future Scope of the project

Enhanced Customization Options: Allow users to customize QR codes further by adding logos, changing colors, or embedding additional information such as images or contact details.

Integration with Other Systems: Integrate the QR code generator with other systems and platforms such as e-commerce websites, inventory management systems, or event management software to generate QR codes automatically for products, assets, or tickets.

Analytics and Tracking: Implement analytics and tracking features to monitor QR code usage, such as the number of scans, location data, and time of scan. This data can provide valuable insights for marketing campaigns, inventory management, or event attendance tracking.

Dynamic QR Codes: Enable the generation of dynamic QR codes that can be updated with new information without changing the QR code itself. This could be useful for promotions, coupons, or real-time updates.

Security Enhancements: Implement security measures to prevent QR code tampering or misuse, such as encryption, digital signatures, or access controls.

Mobile App Integration: Develop a mobile app that allows users to generate QR codes on the go, scan existing QR codes, and access additional features such as augmented reality or location-based services.

Multi-language Support: Offer support for multiple languages to cater to a global audience and ensure accessibility for users from different regions.

Offline Capabilities: Develop offline capabilities so users can generate QR codes and access previously generated codes without an internet connection. This could be useful in areas with limited connectivity or for applications where internet access is not guaranteed.

APIs and SDKs: Provide APIs and SDKs to allow third-party developers to integrate the QR code generator into their own applications and services, expanding the reach and potential use cases of the project.

Accessibility Features: Ensure that the QR code generator is accessible to users with disabilities by implementing features such as screen reader support, keyboard navigation, and alternative text descriptions for generated QR codes.

Cross-platform Compatibility: Ensure compatibility with a wide range of devices and platforms, including mobile devices, tablets, desktop computers, and various operating systems.

Machine Learning Integration: Explore the use of machine learning algorithms to optimize QR code generation, improve error correction capabilities, or provide personalized recommendations based on user behavior and preferences.

References

- **CSS tutorial. (n.d.).** W3Schools Online Web Tutorials.
<https://www.w3schools.com/css/>
- **DevDocs** — JavaScript documentation. (n.d.). DevDocs API Documentation.
<https://devdocs.io/javascript/>
- **HTML tutorial. (n.d.).** W3Schools Online Web Tutorials.
<https://www.w3schools.com/html/>
- **(n.d.). npm Docs.**
<https://docs.npmjs.com/>
- **JavaScript..** MDN Web Docs.
<https://developer.mozilla.org/enUS/docs/Web/JavaScript>
- **React + Vita :**
<https://react.dev/reference/react>
- **Node Js**
<https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- **Express**
<https://expressjs.com/en/5x/api.html>
- **MongoDB Atlas**
<https://www.mongodb.com/docs/>