Barcode & qrcode generator

**project group:**

*Sardar thafzil ahamed (sarda2t)*

*Padmaraju yaswini (padma1y)*

*Sai Kiran Debbadi (debba1s)*

*Sri Srujani kandula (kandu3s)*

# introduction

The proposed project is a **web-based application** to generate barcodes and QR codes for the given text. The application will have a front-end with **Angular** and a back-end built with **Python**.

We are thinking to have the user interface with two pages, one for generating barcodes and the other for generating QR codes. The application will also include the option to download and share the generated barcodes or QR codes.

# Objectives:

1. To build a web application for generating barcodes and QR codes for the given text.

2. To design and develop a responsive UI for the application using Angular.

3. To implement a Python back-end service that will handle the generation of barcodes and QR codes.

4. To allow users to download and share the generated barcodes or QR codes from the corresponding pages.

# Features:

1. Two pages for barcode and QR code generation.

2. Options to input text for generating the barcode or QR code.

3. Options to download and share the generated barcode or QR code.

4. A Python-based back-end service for generating the barcodes and QR codes.

5. User-friendly and responsive UI using Angular.

6. Real-time preview of the generated barcode or QR code.

7. Error handling for incorrect or invalid input.

# Technologies Used:

Angular: Angular is a popular front-end web application framework that allows developers to create robust and responsive user interfaces.

Python: Python is a widely used programming language that will be used for the back-end service to generate barcodes and QR codes.

JavaScript & TypeScript: JavaScript and TypeScript will be used for front-end functionalities such as real-time preview of the generated barcode or QR code.

# Methodology:

We are thinking to divide our project into few phases in the given timeline **(3 weeks)**

1. Requirements gathering and analysis: The first phase will involve gathering and analyzing the requirements for the application. This will involve identifying the features, functionality, and performance requirements of the application.

2. Designing the user interface using Angular: Once the requirements are done, the second phase will be to design the front-end using Angular framework. This will involve designing the pages for barcode and QR code generation, as well as incorporating features such as real-time preview, input validation, adding download & share buttons, and error handling.

3. Setting up the back-end service in Python: The third phase will be to set up the back-end service in Python. This will involve designing and implementing the algorithms for generating barcodes and QR codes by importing few libraries. The back-end service will also be responsible for handling user input and providing the generated code images to the front-end. Along with this, we will test the API using HTTPie or Postman.

4. Integration of Full-Stack web app: This phase will involve integrating the Angular front-end with the Python back-end service. The two pages for barcode and QR code generation will be implemented, along with features such as input validation, error handling, and real-time preview.

5. Dockerize the web application: The next phase will be to dockerize the application. This will involve creating a Dockerfile that specifies the application's dependencies, configuration, and runtime environment. The Dockerfile will be used to build a Docker image that contains the application and its dependencies.

6. Deploying the application: The final phase will be to deploy the application in a Docker container which will be a local and cloud (Google Cloud Platform) container. The container will be configured to run the application, and any necessary networking, storage, port, and volume configurations will be implemented.

# Conclusion:

This Barcode and QR Code Generator Application will be designed, developed, and deployed in a Docker container. The application will have a user-friendly and responsive user interface built using Angular, and a Python-based back-end service that generates barcodes and QR codes. The application will also be deployed in a Google Cloud Platform container, which will provide a lightweight and portable runtime environment that can be easily scaled and managed.