

QR 코드 생성기 프로젝트 제안서

오픈소스SW활용 1분반

팀장 : 장천명(32203919)

팀원 : 김준섭(32217184), 이근태(32203052)

목차

1. 프로젝트 개요
2. 기능 요구사항
3. 비기능 요구사항
4. 시스템 요구사항
5. 테스트 요구사항
6. 시나리오
7. UML 다이어그램
8. 활동 다이어그램
9. UI Sketch
10. 개발 전략
11. 테스트 계획
12. 참고 문헌

1. 프로젝트 개요

본 프로젝트는 사용자가 입력한 텍스트(문자열) 또는 URL을 QR 코드로 변환하여 시각적으로 제공하고, QR 코드를 다운로드할 수 있는 웹 애플리케이션입니다. 간단하고 직관적인 사용자 인터페이스와 신뢰성 높은 동작을 목표로 합니다.

2. 기능 요구사항

2.1 기본 기능

1. 텍스트 또는 URL 입력 기능:

- 사용자는 텍스트 또는 URL을 입력할 수 있어야 합니다.

2. QR 코드 생성:

- 입력된 데이터를 기반으로 QR 코드를 생성하여 화면에 출력합니다.

3. QR 코드 다운로드:

- 사용자는 생성된 QR 코드를 PNG 이미지 형식으로 다운로드할 수 있어야 합니다.

2.2 추가 기능

1. QR 코드 크기 조정:

- 사용자가 원하는 크기로 QR 코드를 생성할 수 있어야 합니다.

2. 색상 커스터마이징:

- QR 코드의 배경색 및 전경색을 사용자 지정 가능.

3. 다중 QR 코드 생성:

- 여러 개의 텍스트를 입력하여 다수의 QR 코드를 동시에 생성.

4. 템플릿 저장 및 재사용:

- QR 코드 설정(크기, 색상 등)을 저장하고, 이후 재사용 가능.

3. 비기능 요구사항

1. 반응형 디자인:

- 모바일, 태블릿, 데스크톱 환경에서 QR 코드 생성기가 정상적으로 동작해야 합니다.

2. 성능 요구사항:

- QR 코드 생성 속도는 입력 후 1초 이내여야 합니다.

3. 안정성:

- 애플리케이션이 중단되지 않아야 합니다.

4. 브라우저 호환성:

- 최신 버전의 주요 브라우저(Chrome, Firefox, Edge, Safari)에서 동작해야 합니다.

4. 시스템 요구사항

1. 클라이언트:

- 브라우저를 통해 실행 가능한 정적 웹 애플리케이션.
- HTML, CSS, JavaScript로 작성.

2. 라이브러리:

- QRCode.js 라이브러리를 사용하여 QR 코드 생성.

3. 호스팅:

- GitHub Pages, Replit, 또는 기타 정적 웹 호스팅 플랫폼에서 배포 가능.

4. 저장소:

- 외부 데이터베이스는 사용하지 않으며, 모든 데이터는 클라이언트에서 처리.

5. 테스트 요구사항

1. 기능 테스트:

- QR 코드 생성, 다운로드, 유효성 검사 기능의 정상 동작 여부 확인.

2. 브라우저 호환성 테스트:

- Chrome, Firefox, Edge, Safari에서의 동작 확인.

3. 반응형 디자인 테스트:

- 모바일, 태블릿, 데스크톱 환경에서 레이아웃이 깨지지 않는지 확인.

4. 성능 테스트:

- 대량의 입력값 또는 복잡한 URL 처리 시 성능 확인.

6. 시나리오

사용자가 시스템과 상호작용하는 모든 주요 흐름을 포함하고 있으며, 정상 동작과 예외 상황을 모두 다룹니다.

1. 주요 시나리오

1.1 시나리오 1: QR 코드 생성

1. 사용자는 브라우저에서 QR 코드 생성기 웹 애플리케이션에 접속합니다.
2. '텍스트 또는 URL 입력' 필드에 텍스트 또는 URL(예: <https://example.com>)을 입력합니다.
3. 'QR 코드 생성' 버튼을 클릭합니다.
4. 시스템은 입력값을 확인하고 유효하면 QR 코드를 생성하여 화면에 표시합니다.
5. QR 코드 아래에 'QR 코드 다운로드' 버튼이 활성화됩니다.

1.2 시나리오 2: 유효하지 않은 입력값 처리

1. 사용자는 입력 필드에 아무 값도 입력하지 않고 'QR 코드 생성' 버튼을 클릭합니다.
2. 시스템은 입력값이 비어 있음을 확인하고 경고 메시지를 표시합니다:
'텍스트 또는 URL을 입력해주세요.'
3. QR 코드는 생성되지 않으며, 기존 화면 상태가 유지됩니다.

1.3 시나리오 3: QR 코드 다운로드

1. 사용자는 텍스트를 입력하고 QR 코드를 생성합니다.
2. 'QR 코드 다운로드' 버튼을 클릭합니다.
3. 시스템은 QR 코드를 PNG 이미지로 변환하여 사용자가 다운로드할 수 있도록 제공합니다.
4. QR 코드는 'qr-code.png'라는 이름으로 저장됩니다.

1.4 시나리오 4: 모바일 환경에서 사용

1. 사용자는 스마트폰 브라우저에서 QR 코드 생성기에 접속합니다.
2. 텍스트를 입력하고 'QR 코드 생성' 버튼을 클릭합니다.
3. 시스템은 모바일 친화적인 레이아웃으로 QR 코드를 화면에 표시합니다.
4. 사용자는 QR 코드를 다운로드하거나 화면을 캡처하여 저장합니다.

2. 예외 처리 시나리오

2.1 예외 1: 입력값이 너무 길 경우

1. 사용자는 500자 이상의 텍스트를 입력하고 'QR 코드 생성' 버튼을 클릭합니다.
2. 시스템은 입력값이 너무 길다고 판단하여 경고 메시지를 표시합니다:
'입력값이 너무 길어 일부 데이터가 손실될 수 있습니다.'
3. QR 코드는 가능한 데이터만 포함하여 생성됩니다.

3. 사용자 시나리오 목적

1. 사용자가 시스템을 쉽게 이해하고 사용할 수 있도록 도움을 제공합니다.
2. 개발 및 테스트 단계에서 주요 동작 흐름과 예외 상황을 검증합니다.
3. 사용자 경험 개선을 위한 피드백 및 문제 해결의 근거를 제공합니다.

7. UML 다이어그램

1. 주요 클래스

1. MainApp:

- 애플리케이션의 메인 로직을 관리합니다.
- UIHandler, QRCodeGenerator, FileHandler 객체를 포함합니다.

2. UIHandler:

- 사용자 인터페이스 요소와 이벤트를 처리합니다.
- 입력 필드, 버튼, QR 코드 출력 영역 등을 관리합니다.

3. QRCodeGenerator:

- QR 코드를 생성하는 핵심 로직을 포함합니다.
- QRCode.js 라이브러리를 사용하여 QR 코드를 생성합니다.

4. FileHandler:

- QR 코드를 이미지 파일로 저장하거나 다운로드를 처리합니다.
- PNG 형식으로 저장 기능을 제공합니다.

2. UML 클래스 다이어그램

다음은 QR 코드 생성기의 클래스 다이어그램 구성 요소입니다:

1. MainApp:

- uiHandler: UIHandler
- qrCodeGenerator: QRCodeGenerator
- fileHandler: FileHandler
- 메소드: initApp(): void

2. UIHandler:

- 속성: inputField: string, generateButton: Button, downloadButton: Button, outputDiv: Div
- 메소드: captureInput(): string, displayQRCode(qrCode: Canvas): void, showError(message: string): void

3. QRCodeGenerator:

- 속성: library: QRCode.js
- 메소드: generateQRCode(data: string, options: Object): Canvas

4. FileHandler:

- 속성: fileFormat: string
- 메소드: saveAsImage(canvas: Canvas): void

8. 활동 다이어그램

[시작]

|

[사용자가 텍스트 또는 URL 입력]

|

[QR 코드 생성 버튼 클릭]

|

[QR 코드 생성 요청]

|

[QR 코드 생성 성공]

|

[QR 코드 화면에 표시]

|

[사용자가 QR 코드 다운로드 클릭]

|

[QR 코드 PNG 파일로 저장]

|

[종료]

1. 단계별 설명

1.1 시작

사용자가 QR 코드 생성기 웹 애플리케이션에 접속하여 시스템과 상호작용을 시작합니다.

1.2 텍스트 또는 URL 입력

사용자는 입력 필드에 텍스트(예: 문자열 또는 URL)를 입력합니다.

1.3 QR 코드 생성 버튼 클릭

사용자가 'QR 코드 생성' 버튼을 클릭하여 시스템에 QR 코드 생성을 요청합니다.

1.4 QR 코드 생성 요청

입력값이 유효하면 QR 코드 생성 로직을 호출합니다(QRCode.js 사용).

1.5 QR 코드 생성 성공

QR 코드를 생성하고 이를 화면에 표시합니다.

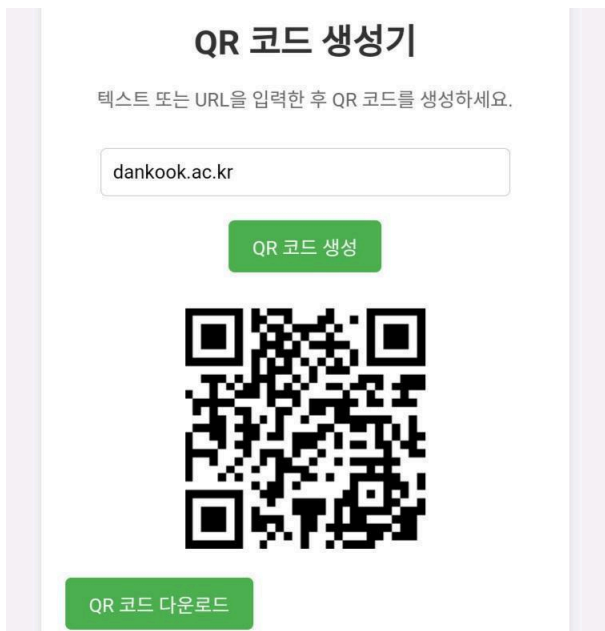
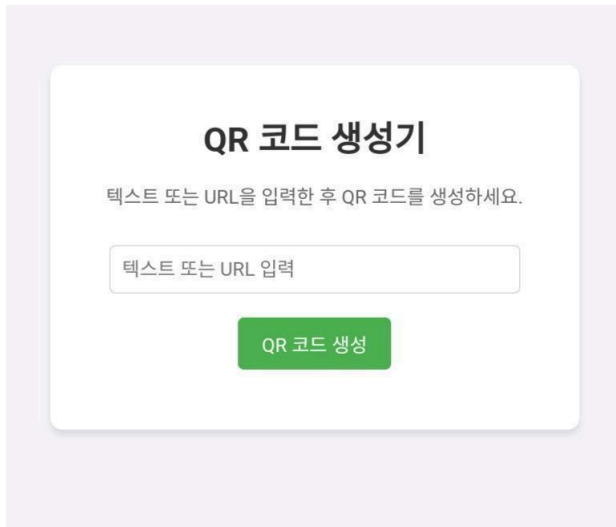
1.6 QR 코드 다운로드

사용자가 'QR 코드 다운로드' 버튼을 클릭하면 QR 코드는 PNG 파일 형식으로 저장됩니다.

1.7 종료

모든 작업이 완료되고 시스템이 대기 상태로 돌아갑니다.

9. UI Sketch



10. 개발 전략

이 애플리케이션의 개발 전략, 사용 언어, API 및 프레임워크, 개발 도구, 개발 일정을 포함합니다.

1. 단계별 개발 계획

1. **기본 기능 구현**:

- 입력 필드, QR 코드 생성 버튼, 결과 출력 영역을 포함한 UI 설계.
- QRCode.js 라이브러리를 활용하여 QR 코드 생성 기능 구현.

2. **유효성 검사 추가**:

- 입력값이 비어 있는 경우 경고 메시지 출력.
- 잘못된 데이터 형식(URL 등)에 대한 에러 처리.

3. ****QR 코드 다운로드 기능 추가****:

- QR 코드를 PNG 이미지 형식으로 저장할 수 있는 기능 구현.

4. ****반응형 UI 설계****:

- 모바일, 태블릿, 데스크톱 환경에서 동작하도록 레이아웃 최적화.

5. ****테스트 및 디버깅****:

- 기능별 테스트를 수행하여 오류를 수정하고 성능을 최적화.
- 주요 브라우저와 디바이스에서 호환성 확인.

2. 사용 언어 및 기술

1. ****프로그래밍 언어****:

- HTML: 웹 애플리케이션의 구조 정의.
- CSS: 사용자 인터페이스의 스타일링 및 레이아웃.
- JavaScript: 동적 동작 구현 및 QR 코드 생성 로직.

2. ****API 및 프레임워크****:

- QRCode.js: QR 코드 생성 로직 구현.
- 선택적으로 외부 QR 코드 API를 연동하여 확장 가능.

3. ****개발 도구****:

- 코드 에디터: Visual Studio Code.
- 브라우저 개발자 도구: Chrome DevTools, Firefox DevTools.
- 버전 관리: Git 및 GitHub.
- 배포 플랫폼: GitHub Pages, Replit.

3. 개발 일정

총 개발 기간: 5일

1. ****Day 1****: 프로젝트 초기화 및 기본 UI 설계.

- 입력 필드, 버튼, 결과 출력 영역 설계.
- QRCode.js 라이브러리 추가 및 테스트.

2. ****Day 2****: 기본 기능 구현.

- QR 코드 생성 기능 완성.
- 입력값 유효성 검사 로직 추가.

3. ****Day 3****: QR 코드 다운로드 기능 구현.

- 생성된 QR 코드를 PNG 형식으로 저장할 수 있는 기능 개발.

4. ****Day 4****: 반응형 UI 설계 및 디버깅.

- 모바일 및 태블릿 환경에서의 레이아웃 최적화.

- 다양한 브라우저에서 테스트.

5. ****Day 5****: 최종 테스트 및 배포.

- 주요 기능 테스트.
- GitHub Pages 또는 Replit을 사용하여 배포.

4. 기대 효과

1. ****사용성 증대****:

- 간단하고 직관적인 QR 코드 생성기를 통해 사용자 만족도 향상

2. ****확장성****:

- 추가 기능 구현 및 상업적 활용 가능성을 탐구할 수 있는 기반 마련.

11. 테스트 계획

주요 기능에 대한 테스트 케이스, 테스트 방법, 그리고 통과 여부를 판단할 수 있는 근거를 포함합니다.

1. 테스트 케이스

테스트 ID	테스트 시나리오	예상 결과	통과 판단 근거
TC-001	사용자가 텍스트를 입력하고 'QR 코드 생성' 버튼을 클릭.	QR 코드가 생성되고 화면에 표시됨.	화면에 QR 코드가 올바르게 표시되었는지 확인.
TC-002	입력값 없이 'QR 코드 생성' 버튼을 클릭.	오류 메시지가 표시됨.	오류 메시지가 예상된 텍스트('텍스트 또는 URL을 입력해주세요.')와 일치.
TC-003	QR 코드 생성 후 'QR 코드 다운로드' 버튼을 클릭.	QR 코드가 PNG 이미지 형식으로 다운로드됨.	다운로드된 파일이 'qr-code.png' 이름이며, QR 코드가 올바르게 표시됨.
TC-004	500자 이상의 텍스트를 입력하고 QR 코드 생성.	경고 메시지 표시 및 가능한 데이터로 QR 코드 생성.	경고 메시지와 생성된 QR 코드의 데이터 정확성 확인.
TC-005	모바일 브라우저에서 QR 코드 생성.	반응형 UI에서 QR 코드가 올바르게 표시됨.	모바일 화면에서 QR 코드와 UI가 올바르게 표시됨.

2. 테스트 방법

1. **기능 테스트**:

- 주요 기능(입력, QR 코드 생성, 다운로드)에 대한 동작 확인.

2. **유효성 검사**:

- 입력값 없이 버튼 클릭 시 오류 메시지가 올바르게 표시되는지 확인.

3. **호환성 테스트**:

- 주요 브라우저(Chrome, Firefox, Edge, Safari)와 모바일 환경에서 동작 확인.

4. **성능 테스트**:

- 대량의 입력 데이터 또는 긴 URL 처리 시 성능 확인.

5. **디자인 테스트**:

- 반응형 UI가 모바일 및 태블릿 화면에서 정상적으로 동작하는지 확인.

3. 통과 판단 기준

테스트는 다음 기준에 따라 통과 여부를 판단합니다:

1. 각 기능이 명세된 동작을 정확히 수행할 것.
2. 오류 상황에서 적절한 경고 메시지가 표시될 것.
3. 생성된 QR 코드가 정확하고 다운로드 가능한 상태일 것.
4. 다양한 브라우저와 디바이스에서 동일하게 동작할 것.
5. 반응형 UI가 모든 화면 크기에서 올바르게 표시될 것.

12. 참고 문헌

1. 주요 참고 자료

1. QRCode.js GitHub Repository:

- URL: <https://github.com/davidshimjs/qrcodejs>
- Description: QR 코드 생성 라이브러리로, 본 프로젝트에서 QR 코드 생성 기능 구현에 활용됨.

2. Mozilla Developer Network (MDN):

- URL: <https://developer.mozilla.org/>
- Description: HTML, CSS, JavaScript에 대한 참고 자료 및 코드 예제를 제공.

3. W3Schools:

- URL: <https://www.w3schools.com/>
- Description: 웹 개발 관련 튜토리얼과 예제 자료 제공.

4. Stack Overflow:

- URL: <https://stackoverflow.com/>
- Description: 개발 중 발생한 문제를 해결하기 위한 질문과 답변 커뮤니티.

5. GitHub Pages Documentation:

- URL: <https://pages.github.com/>
- Description: 프로젝트를 정적 웹 사이트로 배포하기 위한 가이드.

6. Replit Documentation:

- URL: <https://docs.replit.com/>
- Description: Replit을 활용하여 프로젝트 개발 및 배포하는 방법 제공.