

QR 코드 생성기 최종보고서

오픈소스 SW 활용 1 분반

팀장 : 장천명(32203919)

팀원 : 김준섭(32217184), 이근태(32203052)

목차

1. 프로젝트 개요
2. 제공 기능
3. 시나리오
4. 설계 및 구현 내용
5. 테스트 결과
6. 참고 문헌
7. 데모

1. 프로젝트 개요

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

2. 제공 기능

1. 텍스트 또는 URL 입력 기능:

- 사용자는 텍스트 또는 URL 을 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.

3. 시나리오

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.

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 URL 을 입력해주세요.'
3. QR 코드는 생성되지 않으며, 기존 화면 상태가 유지됩니다.

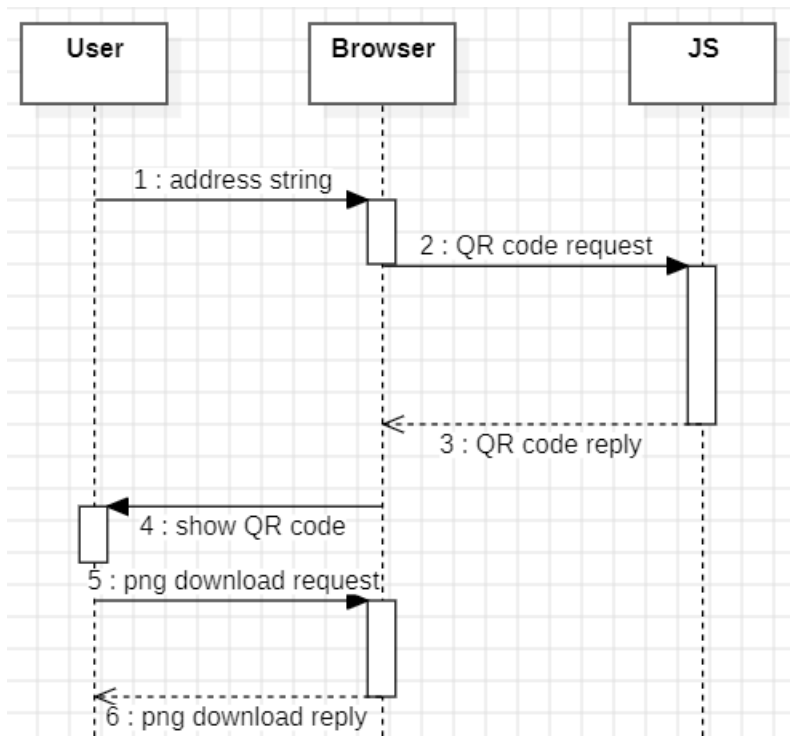
시나리오 3: QR 코드 다운로드

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.

4. 설계 및 구현 내용

4-1. 설계

(1) 시나리오에 따른 시퀀스 다이어그램



(2) 시스템 구조

1. 프론트엔드 : HTML, CSS, JavaScript
2. 라이브러리 : QR 코드 생성을 위해 qrcode.js 사용

(3) 파일 구조

project/

|-- index.html // 웹 페이지 구조 정의

|-- style.css // 사용자 인터페이스 스타일링

|-- script.js // QR 코드 생성 및 다운로드 로직

|-- qrcode.min.js // QR 코드 생성 라이브러리

(4) 주요 클래스 및 함수 설계

1. QRCode 클래스(qrcode.min.js)

- 역할 : QR 코드를 생성하고 렌더링

- 속성

: element : QR 코드가 렌더링될 DOM 요소

: options : QR 코드 설정(텍스트, 크기, 오류 정정 레벨 등)

- 주요 메소드

: makeCode(text) : 입력 텍스트를 QR 코드로 변환

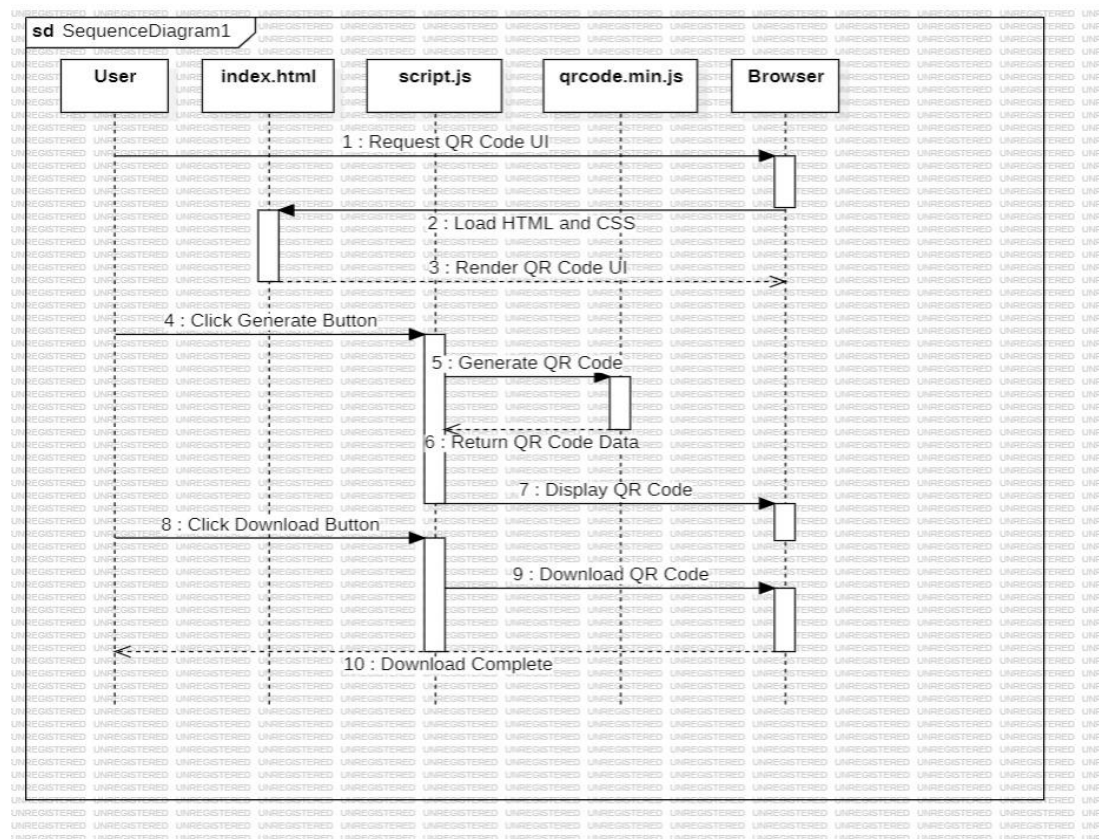
: clear() : 기존 QR 코드를 초기화

: makeImage() : QR 코드를 이미지로 변환

2. generateQRCode() : 사용자가 입력한 데이터를 기반으로 QR 코드를 생성

3. downloadQRCode() : 생성된 QR 코드를 PNG 이미지로 다운로드

(5) 시퀀스 다이어그램



4-2. 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.

3. 개발 도구

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

4-3. 주요 코드

(1) QRcode 클래스(qrcode.js)

(1) makeCode() 메소드

```
QRCode.prototype.makeCode = function (sText) {  
    this._oQRCode = new QRCodeModel(_getTypeNumber(sText, this._htOption.correctLevel), this._htOption.correctLevel);  
    this._oQRCode.addData(sText);  
    this._oQRCode.make();  
    this._el.title = sText;  
    this._oDrawing.draw(this._oQRCode);  
    this.makeImage();  
};
```

(2) clear() 메소드

```
QRCode.prototype.clear = function () {  
    this._oDrawing.clear();  
};
```

(3) makeImage() 메소드

```
QRCode.prototype.makeImage = function () {  
    if (typeof this._oDrawing.makeImage == "function" && (!this._android || this._android >= 3)) {  
        this._oDrawing.makeImage();  
    }  
};
```

(2) generateQRcode() 함수

```
// QR 코드 생성 함수
function generateQRcode() {
  const inputValue = qrInput.value.trim(); // 입력값 가져오기 및 공백 제거

  // 입력값 검증
  if (!inputValue) {
    alert("텍스트 또는 URL을 입력해주세요."); // 입력값 없을 시 경고
    return;
  }

  // 기존 QR 코드 초기화
  qrCodeDiv.innerHTML = "";
  downloadBtn.style.display = "none"; // 초기화 시 다운로드 버튼 숨기기

  // QR 코드 생성
  const qrCode = new QRCode(qrCodeDiv, {
    text: inputValue, // QR 코드에 포함될 데이터
    width: 200, // QR 코드 너비
    height: 200, // QR 코드 높이
    correctLevel: QRCode.CorrectLevel.H,
  });

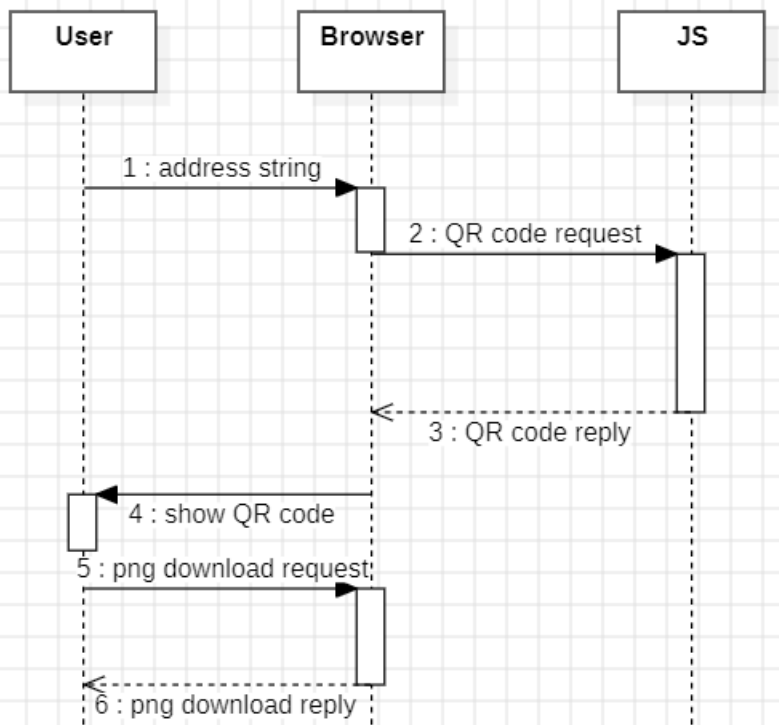
  // 다운로드 버튼 활성화
  setTimeout(() => {
    downloadBtn.style.display = "inline-block"; // 다운로드 버튼 표시
  }, 100);
}
```

(3) downloadQRcode() 함수

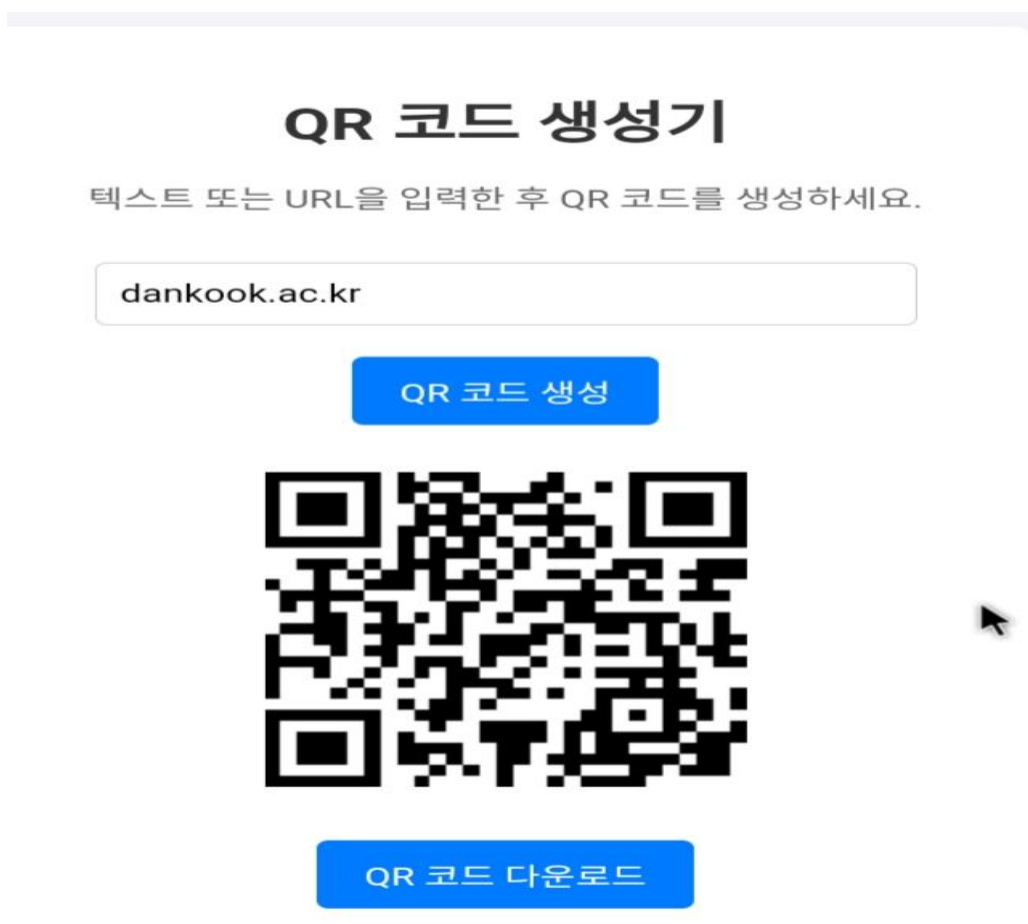
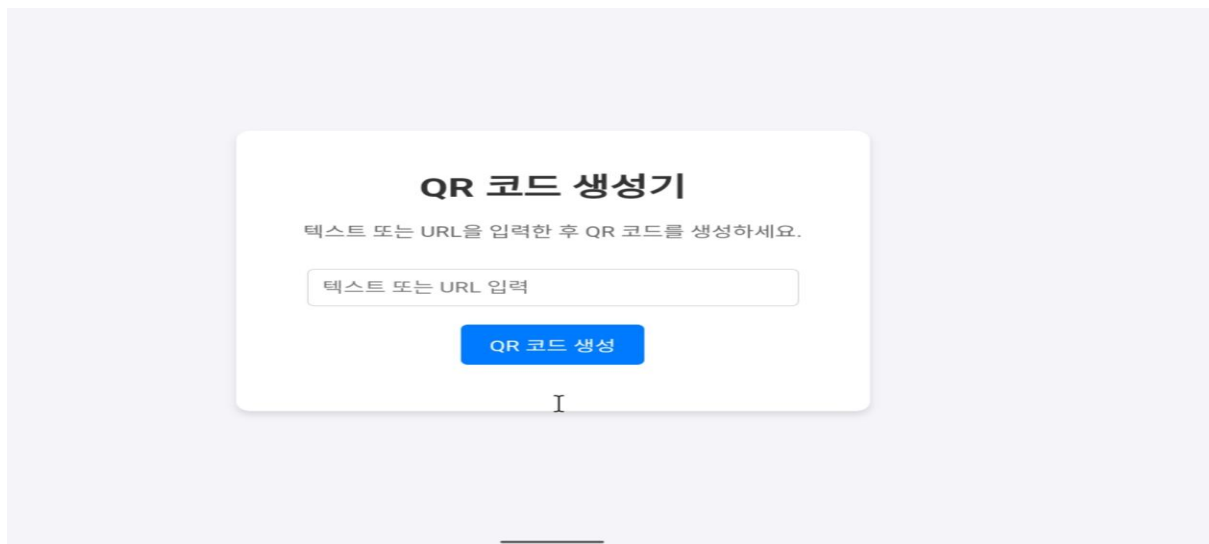
```
// QR 코드 다운로드 함수
function downloadQRcode() {
  const qrCanvas = qrCodeDiv.querySelector("canvas");
  if (qrCanvas) {
    qrCanvas.toBlob((blob) => {
      if (blob) {
        const qrImageURL = URL.createObjectURL(blob); // Blob URL 생성
        console.log("QR Image Blob URL: ", qrImageURL); // 디버그 로그

        // 다운로드 링크 생성 및 클릭 실행
        const link = document.createElement("a");
        link.href = qrImageURL;
        link.download = "qr-code.png"; // 다운로드 파일명 설정
        document.body.appendChild(link);
        link.click(); // 다운로드 강제 실행
        document.body.removeChild(link);
        URL.revokeObjectURL(qrImageURL); // 메모리 해제
      } else {
        alert("QR 코드를 Blob으로 변환하지 못했습니다.");
      }
    }, "image/png");
  } else {
    alert("QR 코드를 생성하지 못했습니다.");
  }
}
```

4-4. 시나리오에 따른 시퀀스 다이어그램



4-5. 구현 결과



5. 테스트 결과

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.

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. 통과 판단 기준

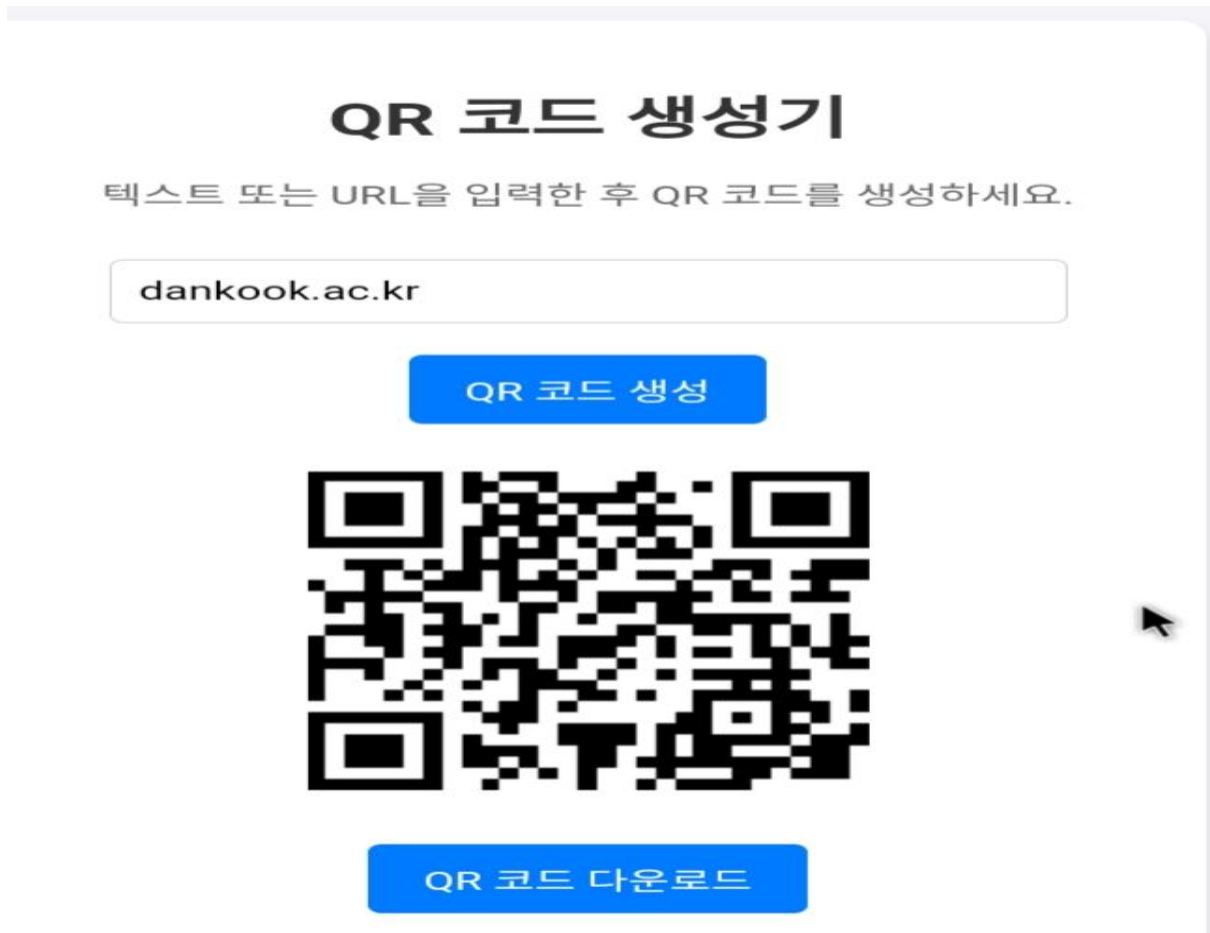
테스트는 다음 기준에 따라 통과 여부를 판단합니다:

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. 테스트 이미지

TC-001

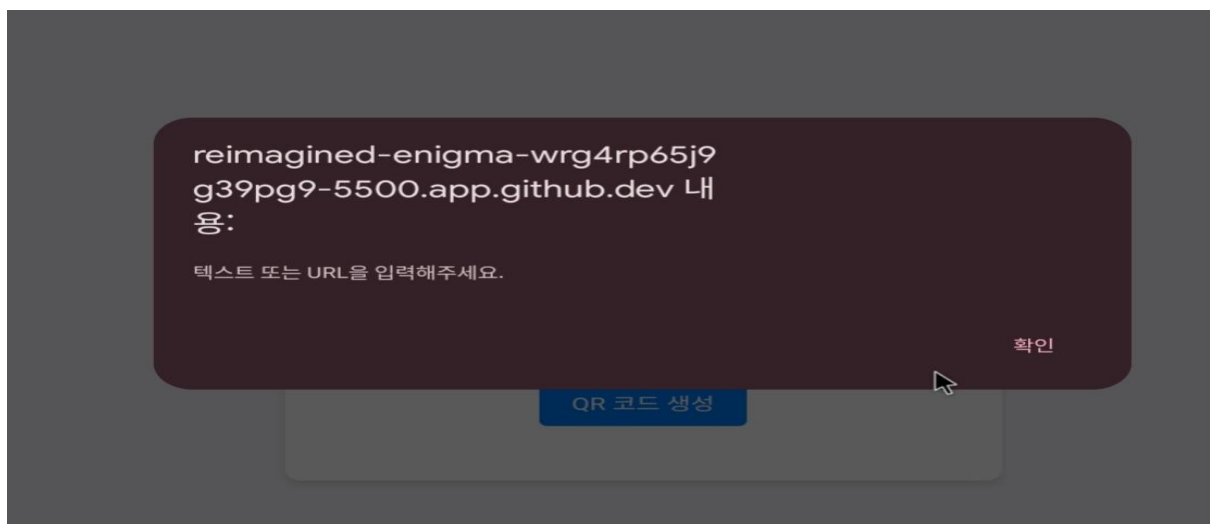
사용자가 텍스트를 입력하고 'QR 코드 생성' 버튼을 클릭,
화면에 QR 코드가 올바르게 표시되었는지 확인



TC-002

입력값 없이 'QR 코드 생성' 버튼을 클릭.

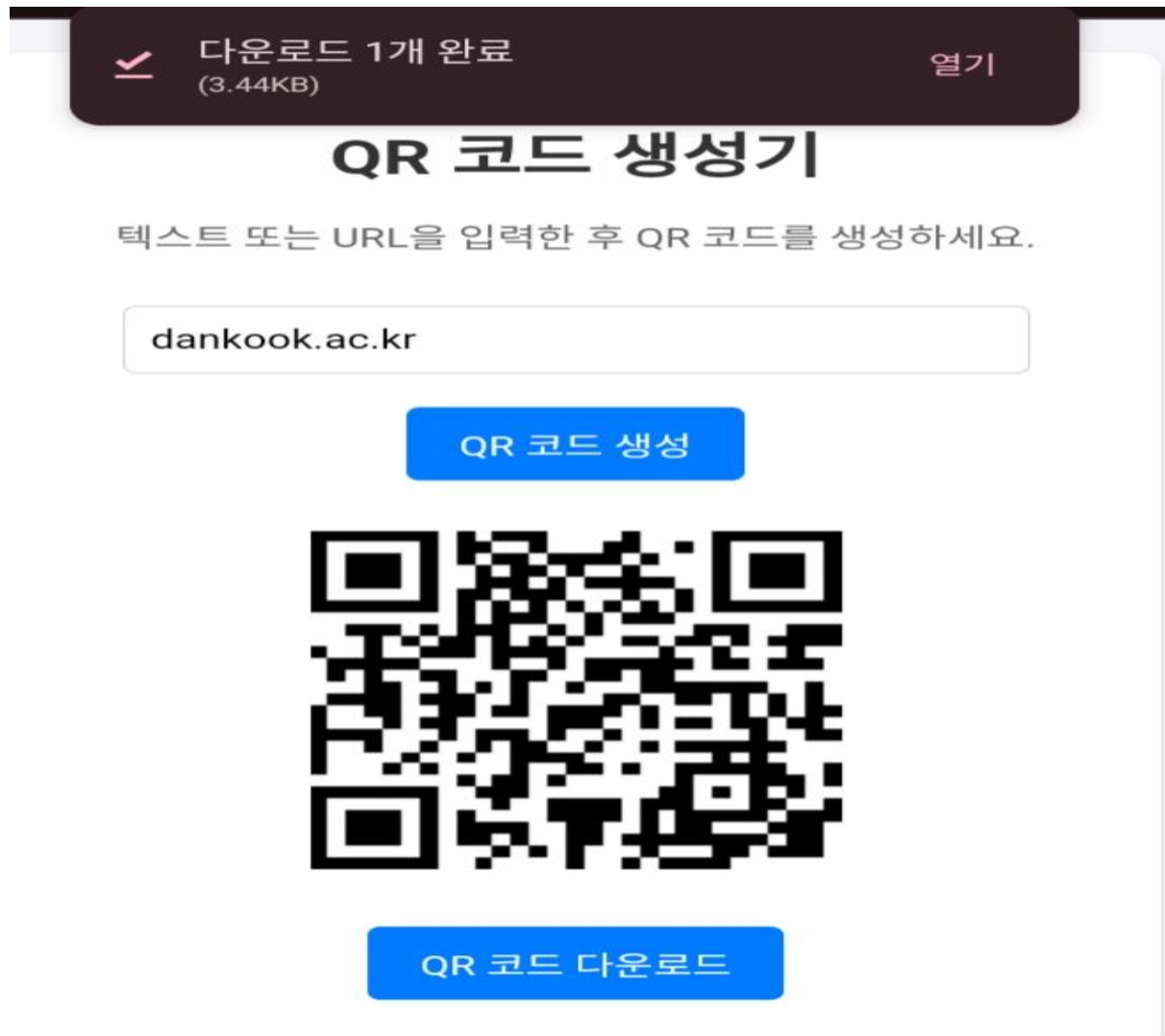
오류 메시지가 예상된 텍스트('텍스트 또는 URL을 입력해주세요.')와 일치



TC-003

QR 코드 생성 후 'QR 코드 다운로드' 버튼을 클릭

다운로드된 파일이 'qr-code.png' 이름이며, QR 코드가 올바르게 표시됨



6. 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, JavaScript 에 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.

7. 데모

https://youtu.be/rW0kGWCIT24?si=zFoojP_9LEJvqqMD