VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE3501 – Information Security Analysis and Audit

J Component Report

A project report titled

Authentication using OTP and QR

By

19BLC1078	A KRISHNA KALYAN
19BLC1135	SWAPNIL BOHIDAR
19BEC1308	UTKARSH MAURYA

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMPUTER ENGINEERING

Submitted to

Vijaya Kumar P

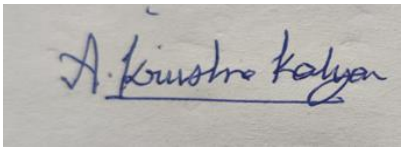
School of Computer Science and Engineering

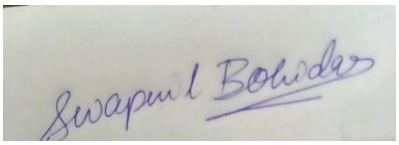
November 2021

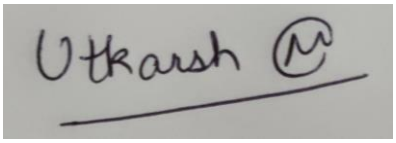
DECLARATION BY THE CANDIDATE

I hereby declare that the report titled "**Authentication using OTP and QR ”** Submitted by me to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of **Vijaya Kumar P, Associate Professor, SCOPE, Vellore Institute of Technology, Chennai.**

Signature of the Candidate







ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Vijayakumar P,** School of Computer Science and Engineering for her consistent encouragement and valuable guidance offered to us throughout the course of the project work.

We are extremely grateful to **Dr. R. Ganesan, Dean,** School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for extending the facilities of the school towards our project and for his unstinting support.

We express our thanks to our **Head of the Department** for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the courses.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

BONAFIDE CERTIFICATE

Certified that this project report entitled "**Authentication using OTP and QR"** is a bona-fide work of **A Krishna Kalyan (19BLC1078), Swapnil Bohidar (19BLC1135), Utkarsh Maurya (19BEC1308) carried** out the "J"-Project work under my supervision and guidance for Subject Code – Subject Name.

Vijaya Kumar P

SCOPE

TABLE OF CONTENTS

Ch. No	Chapter	Page Number
1	Introduction	7
2	Related Work	8
3	TRADITIONAL AUTHENTICATION SYSTEM AND ITS LIMITATIONS	9
4	PROPOSED ALGORITHM	10-11
5	Conclusion	12
6	Appendix 1.Code 2.Implementation	 13-39 40-43
7	References	44

ABSTRACT

- Ensuring the user authentication and the verification of online transactions that are performed on an untrusted computer or device is an important and challenging problem.
- Most people are unaware that scanning an unknown QR code offers serious security concerns. While the QR code itself isn't dangerous, there is no opportunity to evaluate the site it will lead you to such as the case with an email or website
- To help combat security issues around QR code. QR code reader **Norton Snap** verifies the safety of websites before they are allowed to load on your mobile device.
- As we are performing the authentication using QR. The QR only directs to particular URL.
- Before the data encoding process begin, an error correction level is selected to create error correction code words based on the encoded data. By this, error correction code words able to correct the error if QR code reader did not read the data correctly
- Users could generate the QR-Code image for authentication easily through mobile application on their smart devices.
- we analyze the security of our scheme and discuss the mechanisms in the scheme for circumventing a variety of security threats including password stealing, man-in-the-middle and man-in-the-browser attacks.

KEYWORDS: artificial intelligence algorithm; QR image code; image recognition; backpropagation neural networks; two-dimensional code distortion

INTRODUCTION

IMPORTANCE OF TECHNOLOGY

- Smart Phones, greatly expanding in the recent mobile market, are equipped with various features compared to existing feature phones and provide the conveniences to in several ways. The camera, one of the features of a smartphone, creates the digital contents, such photos and videos, and plays a role for the media which transmits information, such as video calls and bar code reader.
- QR-Code recognition is also one of camera features. It contains a variety of information in two-dimensional bar code type in matrix format, and makes it possible to obtain the information by using smart phones.

PROPOSED WORK.

- The user logs in or registers into a website. Now the user is required to open an application on a smartphone which is protected by a pin number and scan the one time QR code that is displayed using the phone's primary camera. The application then communicates with the server through an out of band channel and provides a proof of possession of the device. Here the user acts as the conduit between the authorized device and the authentication entity.

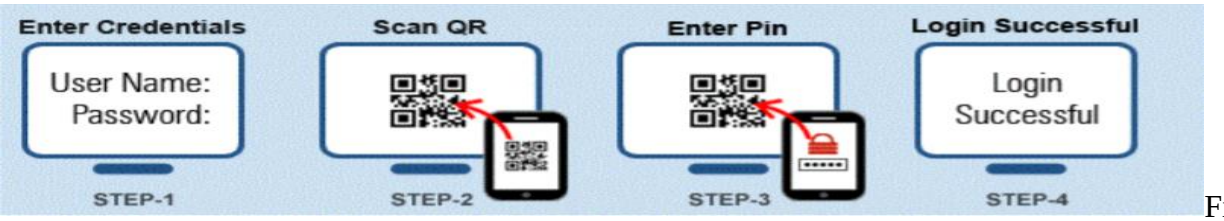


Fig.1: Block diagram of procedure

- The primary aim of this chapter is to verify the effectiveness of the algorithm proposed through the preprocessing of QR code images. The preprocessing process includes grayscale, filtering, binarization, distortion correction, and perspective projection inverse transformation.
- The filtering uses an improved adaptive median filter algorithm to filter the image, and the distortion correction uses a distortion correction method based on the BP neural network.

RELATED WORK

- Hongyu L, Hui C, Ying W, Yong C, Wei Y. Prediction of two-dimensional topography of laser cladding based on neural network. Int J Mod Phys B.2019;19:2–25.
- Rathee G, Sharma A, Saini H, Kumar R, Iqbal R. A hybrid framework for multimedia data processing in IoT-healthcare using blockchain technology. Multimed Tools Appl. 2019;19:1–23.
- Rathee G, Sharma A, Kumar R, Iqbal R. A secure communicating things network framework for industrial IoT using blockchain technology. Ad Hoc Netw.
- Frankovský P, Pástor M, Dominik L, Kicko M, Trebuňa P, Hroncová D, et al. Wheeled mobile robot in structured environment. In 2018 ELEKTRO. IEEE; 2018 May. p. 1–5.

TRADITIONAL AUTHENTICATION SYSTEM AND ITS LIMITATIONS

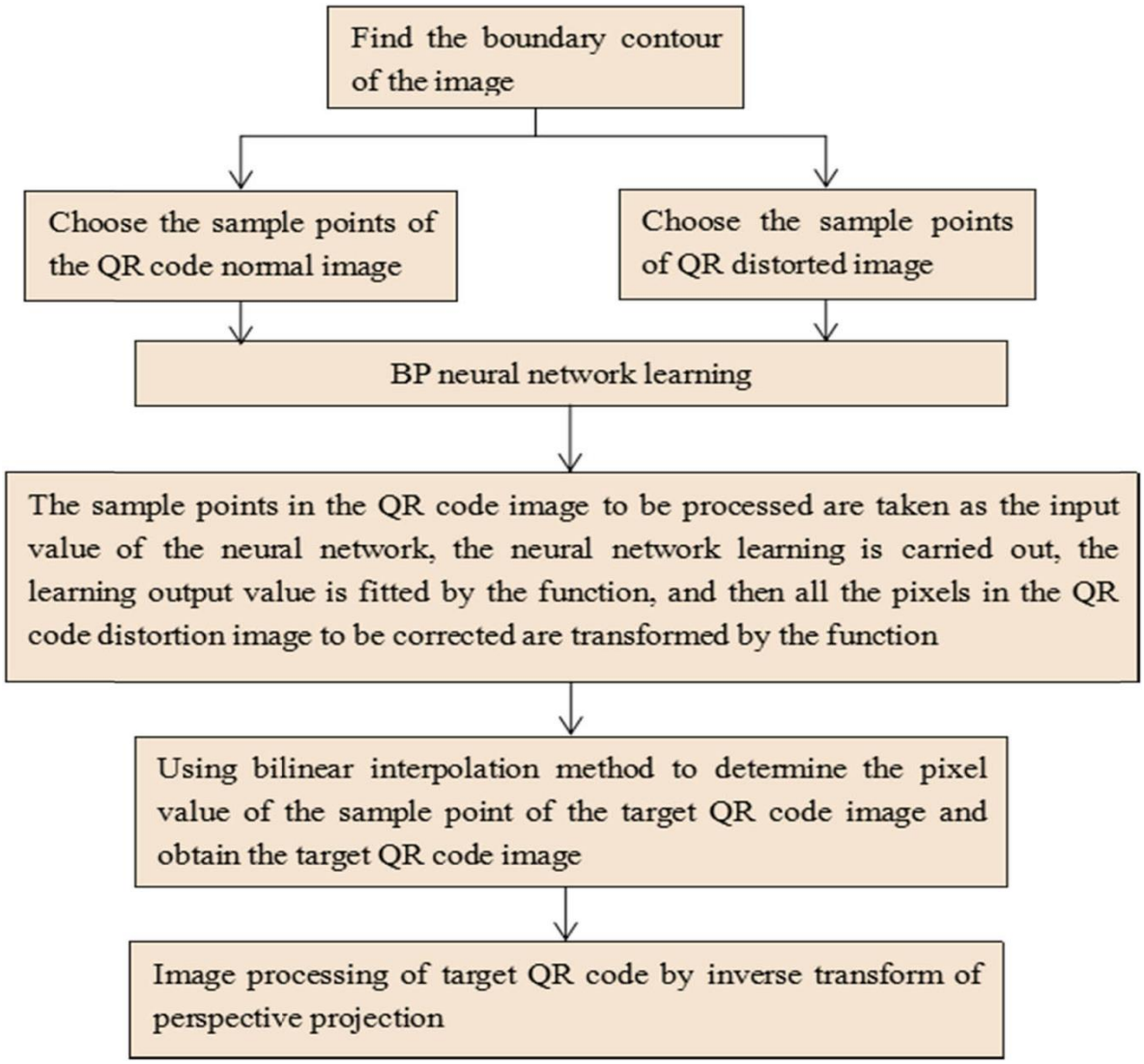
- Password-based authentication.
  - ❖ Passwords are prone to **phishing** attacks and bad hygiene that weakens effectiveness. An average person has about 25 different online accounts, but only **54%** of users use different passwords across their accounts.
- Multi-factor authentication.
  - ❖ MFA may be a good defense against **most account hacks**, but it has its own pitfalls. People may lose their phones or SIM cards and not be able to generate an authentication code.
- Certificate-based authentication.
  - ❖ While the idea of digital certificates is to block outsiders from intercepting your messages, the system is not an infallible one. In 2011, for example, a Dutch digital certificate authority called **DigiNotar**was compromised by hackers.

PROPOSED ALGORITHM

- The basic idea of using the BP neural network to process QR code images is to find the polynomial relationship between these pixel coordinates through the self-learning ability of the BP neural network
- First, consider the point coordinate set on the distorted QR code image as the input layer node of the neural network, and the corresponding output layer node is the point coordinate set on the standard image corresponding to the distorted QR code image.
- The same operation is taken for multiple images, thus forming a set of learning sample data sets. Through the learning of the learning sample set, the neural network forms a distortion mode from point coordinates to point coordinates.
- Snap2pass, and its extension Snap2pay, is a QR code-based approach which requires a system to have an active connection.

To login to a website, the server sends a QR code, which encodes a crypto-graphic challenge, to the browser. The user is to take a picture of the QR code.

Flow Chart


Find the boundary contour of the image
Choose the sample points of the QR code normal image
Choose the sample points of QR distorted image
BP neural network learning
The sample points in the QR code image to be processed are taken as the input value of the neural network, the neural network learning is carried out, the learning output value is fitted by the function, and then all the pixels in the QR code distortion image to be corrected are transformed by the function
Using bilinear interpolation method to determine the pixel value of the sample point of the target QR code image and obtain the target QR code image
Image processing of target QR code by inverse transform of perspective projection

CONCLUSION

This idea can be implemented to achieve security to a great extent in applications such as Net Banking, Online Shopping, detecting Counterfeit products etc. The usage of true random numbers in the generation of QR codes and OTP itself makes it very unique and secure.

The highlight of this technique is that there is no necessity for carrying external hardware such as tokens and smartcards. These features make it a very attractive option for second level authentication in future projects.

APPENDIX

Code /Implementation

1. Index.js

```
import React from “react”
import ReactDOM from “react-dom”
import App from “./components/App”
import “bootstrap/dist/css/bootstrap.min.css”

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById(“root”)
)
```

2. Login.js

```
import React, { useRef, useState } from "react"
import { Form, Button, Card, Alert } from "react-bootstrap"
import { useAuth } from "../contexts/AuthContext"
import { Link, useHistory } from "react-router-dom"

export default function Login() {
  const emailRef = useRef()
  const passwordRef = useRef()
  const { login } = useAuth()
  const [error, setError] = useState("")
  const [loading, setLoading] = useState(false)
  const history = useHistory()

  async function handleSubmit(e) {
    e.preventDefault()

    try {
      setError("")
      setLoading(true)
      await login(emailRef.current.value, passwordRef.current.value)
      history.push("/")
    } catch {
      setError("Failed to log in")
    }

    setLoading(false)
  }
```

```
  return (
    <>
      <Card>
        <Card.Body>
          <h2 className="text-center mb-4">Log In</h2>
          {error && <Alert variant="danger">{error}</Alert>}
          <Form onSubmit={handleSubmit}>
            <Form.Group id="email">
              <Form.Label>Email</Form.Label>
              <Form.Control type="email" ref={emailRef} required />
            </Form.Group>
            <Form.Group id="password">
              <Form.Label>Password</Form.Label>
              <Form.Control type="password" ref={passwordRef} required />
            </Form.Group>
            <Button disabled={loading} className="w-100" type="submit">
              Log In
            </Button>
          </Form>
          <div className="w-100 text-center mt-3">
            <Link to="/forgot-password">Forgot Password?</Link>
          </div>
        </Card.Body>
      </Card>
      <div className="w-100 text-center mt-2">
        Need an account? <Link to="/signup">Sign Up</Link>
      </div>
    </>
  )
}
```

3. Signup.js

```
import React, { useRef, useState } from "react"
import { Form, Button, Card, Alert } from "react-bootstrap"
import { useAuth } from "../contexts/AuthContext"
import { Link, useHistory } from "react-router-dom"

export default function Signup() {
  const emailRef = useRef()
  const passwordRef = useRef()
  const passwordConfirmRef = useRef()
  const { signup } = useAuth()
  const [error, setError] = useState("")
  const [loading, setLoading] = useState(false)
  const history = useHistory()

  async function handleSubmit(e) {
    e.preventDefault()

    if (passwordRef.current.value !== passwordConfirmRef.current.value) {
      return setError("Passwords do not match")
    }

    try {
      setError("")
      setLoading(true)
      await signup(emailRef.current.value, passwordRef.current.value)
      history.push("/")
    } catch {
      setError("Failed to create an account")
    }

    setLoading(false)
  }

  return (
    <>
      <Card>
        <Card.Body>
          <h2 className="text-center mb-4">Sign Up</h2>
          {error && <Alert variant="danger">{error}</Alert>}
          <Form onSubmit={handleSubmit}>
            <Form.Group id="email">
              <Form.Label>Email</Form.Label>
              <Form.Control type="email" ref={emailRef} required />
            </Form.Group>
            <Form.Group id="password">
              <Form.Label>Password</Form.Label>
              <Form.Control type="password" ref={passwordRef} required />
            </Form.Group>
            <Form.Group id="password-confirm">
              <Form.Label>Password Confirmation</Form.Label>
```

```
              <Form.Control type="password" ref={passwordConfirmRef} required />
            </Form.Group>
            <Button disabled={loading} className="w-100" type="submit">
              Sign Up
            </Button>
          </Form>
        </Card.Body>
      </Card>
      <div className="w-100 text-center mt-2">
        Already have an account? <Link to="/login">Log In</Link>
      </div>
    </>
  )
}
```

4. UpdateProfile.js

```
import React, { useRef, useState } from "react"
import { Form, Button, Card, Alert } from "react-bootstrap"
import { useAuth } from "../contexts/AuthContext"
import { Link, useHistory } from "react-router-dom"

export default function UpdateProfile() {
  const emailRef = useRef()
  const passwordRef = useRef()
  const passwordConfirmRef = useRef()
  const { currentUser, updatePassword, updateEmail } = useAuth()
  const [error, setError] = useState("")
  const [loading, setLoading] = useState(false)
  const history = useHistory()

  function handleSubmit(e) {
    e.preventDefault()
    if (passwordRef.current.value !== passwordConfirmRef.current.value) {
      return setError("Passwords do not match")
    }

    const promises = []
    setLoading(true)
    setError("")

    if (emailRef.current.value !== currentUser.email) {
      promises.push(updateEmail(emailRef.current.value))
    }
    if (passwordRef.current.value) {
      promises.push(updatePassword(passwordRef.current.value))
    }
```

```
    Promise.all(promises)
      .then(() => {
        history.push("/")
      })
      .catch(() => {
        setError("Failed to update account")
      })
      .finally(() => {
        setLoading(false)
      })
  }

  return (
    <>
      <Card>
        <Card.Body>
          <h2 className="text-center mb-4">Update Profile</h2>
          {error && <Alert variant="danger">{error}</Alert>}
          <Form onSubmit={handleSubmit}>
            <Form.Group id="email">
              <Form.Label>Email</Form.Label>
              <Form.Control
                type="email"
                ref={emailRef}
                required
                defaultValue={currentUser.email}
              />
            </Form.Group>
            <Form.Group id="password">
              <Form.Label>Password</Form.Label>
              <Form.Control
                type="password"
                ref={passwordRef}
                placeholder="Leave blank to keep the same"
              />
            </Form.Group>
            <Form.Group id="password-confirm">
              <Form.Label>Password Confirmation</Form.Label>
              <Form.Control
                type="password"
                ref={passwordConfirmRef}
                placeholder="Leave blank to keep the same"
              />
            </Form.Group>
            <Button disabled={loading} className="w-100" type="submit">
              Update
            </Button>
          </Form>
        </Card.Body>
      </Card>
      <div className="w-100 text-center mt-2">
```

```
        <Link to="/">Cancel</Link>
      </div>
    </>
  )
}
```

5. PrivateRoute.js

```
import React from "react"
import { Route, Redirect } from "react-router-dom"
import { useAuth } from "../contexts/AuthContext"

export default function PrivateRoute({ component: Component, ...rest }) {
  const { currentUser } = useAuth()

  return (
    <Route
      {...rest}
      render={props => {
        return currentUser ? <Component {...props} /> : <Redirect to="/login" />
      }}
    ></Route>
  )
}
```

6. ForgotPassword.js

```
import React, { useRef, useState } from "react"
import { Form, Button, Card, Alert } from "react-bootstrap"
import { useAuth } from "../contexts/AuthContext"
import { Link } from "react-router-dom"

export default function ForgotPassword() {
  const emailRef = useRef()
  const { resetPassword } = useAuth()
  const [error, setError] = useState("")
  const [message, setMessage] = useState("")
  const [loading, setLoading] = useState(false)

  async function handleSubmit(e) {
    e.preventDefault()

    try {
      setMessage("")
      setError("")
      setLoading(true)
      await resetPassword(emailRef.current.value)
      setMessage("Check your inbox for further instructions")
    } catch {
      setError("Failed to reset password")
    }

    setLoading(false)
  }

  return (
    <>
      <Card>
        <Card.Body>
          <h2 className="text-center mb-4">Password Reset</h2>
          {error && <Alert variant="danger">{error}</Alert>}
          {message && <Alert variant="success">{message}</Alert>}
          <Form onSubmit={handleSubmit}>
            <Form.Group id="email">
              <Form.Label>Email</Form.Label>
              <Form.Control type="email" ref={emailRef} required />
            </Form.Group>
            <Button disabled={loading} className="w-100" type="submit">
              Reset Password
            </Button>
          </Form>
          <div className="w-100 text-center mt-3">
            <Link to="/login">Login</Link>
          </div>
        </Card.Body>
      </Card>
      <div className="w-100 text-center mt-2">
```

```
        Need an account? <Link to="/signup">Sign Up</Link>
      </div>
    </>
  )
}
```

7. Dashboard.js

```
import React, { useState } from "react"
import { Card, Button, Alert } from "react-bootstrap"
import { useAuth } from "../contexts/AuthContext"
import { Link, useHistory } from "react-router-dom"

export default function Dashboard() {
  const [error, setError] = useState("")
  const { currentUser, logout } = useAuth()
  const history = useHistory()

  async function handleLogout() {
    setError("")

    try {
      await logout()
      history.push("/login")
    } catch {
      setError("Failed to log out")
    }
  }

  return (
    <>
      <Card>
        <Card.Body>
          <h2 className="text-center mb-4">Profile</h2>
          {error && <Alert variant="danger">{error}</Alert>}
          <strong>Email:</strong> {currentUser.email}
          <Link to="/update-profile" className="btn btn-primary w-100 mt-3">
            Update Profile
          </Link>
        </Card.Body>
      </Card>
      <div className="w-100 text-center mt-2">
        <Button variant="link" onClick={handleLogout}>
          Log Out
        </Button>
      </div>
    </>
  )
}
```

8. App.js

```
import React from "react"
import Signup from "./Signup"
import { Container } from "react-bootstrap"
import { AuthProvider } from "../contexts/AuthContext"
import { BrowserRouter as Router, Switch, Route } from "react-router-dom"
import Dashboard from "./Dashboard"
import Login from "./Login"
import PrivateRoute from "./PrivateRoute"
import ForgotPassword from "./ForgotPassword"
import UpdateProfile from "./UpdateProfile"

function App() {
  return (
    <Container
      className="d-flex align-items-center justify-content-center"
      style={{ minHeight: "100vh" }}
    >
      <div className="w-100" style={{ maxWidth: "400px" }}>
        <Router>
          <AuthProvider>
            <Switch>
              <PrivateRoute exact path="/" component={Dashboard} />
              <PrivateRoute path="/update-profile" component={UpdateProfile} />
              <Route path="/signup" component={Signup} />
              <Route path="/login" component={Login} />
              <Route path="/forgot-password" component={ForgotPassword} />
            </Switch>
          </AuthProvider>
        </Router>
      </div>
    </Container>
  )
}

export default App
```

For QR authentication

1. bower.json

```
{
  "name": "qrcode.js",
  "version": "0.0.1",
  "homepage": "https://github.com/davidshimjs/qrcodejs",
  "authors": [
    "Sangmin Shim", "Sangmin Shim <ssm0123@gmail.com> (http://jaguarjs.com)"
  ],
  "description": "Cross-browser QRCode generator for javascript",
  "main": "qrcode.js",
  "ignore": [
    "bower_components",
    "node_modules",
    "index.html",
    "index.svg",
    "jquery.min.js",
    "qrcode.min.js"
  ]
}
```

2. index-svg.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
  <head>
    <title>Cross-Browser QRCode generator for Javascript</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="viewport" content="width=device-width,initial-scale=1,user-scalable=no" />
    <script type="text/javascript" src="jquery.min.js"></script>
    <script type="text/javascript" src="qrcode.js"></script>
  </head>
  <body>
  <input id="text" type="text" value="QR Code" style="width:80%" />
  <svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
    <g id="qrcode"/>
  </svg>
  <script type="text/javascript">
  var qrcode = new QRCode(document.getElementById("qrcode"), {
      width : 100,
      height : 100,
      useSVG: true
  });

  function makeCode () {
      var elText = document.getElementById("text");
```

```
        if (!elText.value) {
            alert("Input a text");
            elText.focus();
            return;
        }

        qrcode.makeCode(elText.value);
    }

    makeCode();

    $("#text").
        on("blur", function () {
            makeCode();
        }).
        on("keydown", function (e) {
            if (e.keyCode == 13) {
                makeCode();
            }
        });
    </script>
  </body>
</html>
```

3. index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
<head>
<title>Cross-Browser QRCode generator for Javascript</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="viewport" content="width=device-width,initial-scale=1,user-scalable=no" />
<script type="text/javascript" src="jquery.min.js"></script>
<script type="text/javascript" src="qrcode.js"></script>
</head>
<body>
<input id="text" type="text" value="QR" style="width:80%" /><br />
<div id="qrcode" style="width:100px; height:100px; margin-top:15px;"></div>

<script type="text/javascript">
var qrcode = new QRCode(document.getElementById("qrcode"), {
    width : 100,
    height : 100
});

function makeCode () {
    var elText = document.getElementById("text");

    if (!elText.value) {
```

```
        alert("Input a text");
        elText.focus();
        return;
    }

    qrcode.makeCode(elText.value);
}

makeCode();

$("#text").
    on("blur", function () {
        makeCode();
    }).
    on("keydown", function (e) {
        if (e.keyCode == 13) {
            makeCode();
        }
    });
</script>
</body>
```

4. qrcode.js

```
/**
 * @fileoverview
 * - Using the 'QRCode for Javascript library'
 * - Fixed dataset of 'QRCode for Javascript library' for support full-spec.
 * - this library has no dependencies.
 *
 * @author davidshimjs
 * @see <a href="http://www.d-project.com/" target="_blank">http://www.d-project.com/</a>
 * @see <a href="http://jeromeetienne.github.com/jquery-qrcode/"
target="_blank">http://jeromeetienne.github.com/jquery-qrcode/</a>
 */
var QRCode;

(function () {
    //---------------------------------------------------------------------
    // QRCode for JavaScript
    //
    // Copyright (c) 2009 Kazuhiko Arase
    //
    // URL: http://www.d-project.com/
    //
    // Licensed under the MIT license:
    //   http://www.opensource.org/licenses/mit-license.php
    //
    // The word "QR Code" is registered trademark of
```

```
// DENSO WAVE INCORPORATED
//   http://www.denso-wave.com/qrcode/faqpatent-e.html
//
//---------------------------------------------------------------------
function QR8bitByte(data) {
    this.mode = QRMode.MODE_8BIT_BYTE;
    this.data = data;
    this.parsedData = [];

    // Added to support UTF-8 Characters
    for (var i = 0, l = this.data.length; i < l; i++) {
        var byteArray = [];
        var code = this.data.charCodeAt(i);

        if (code > 0x10000) {
            byteArray[0] = 0xF0 | ((code & 0x1C0000) >>> 18);
            byteArray[1] = 0x80 | ((code & 0x3F000) >>> 12);
            byteArray[2] = 0x80 | ((code & 0xFC0) >>> 6);
            byteArray[3] = 0x80 | (code & 0x3F);
        } else if (code > 0x800) {
            byteArray[0] = 0xE0 | ((code & 0xF000) >>> 12);
            byteArray[1] = 0x80 | ((code & 0xFC0) >>> 6);
            byteArray[2] = 0x80 | (code & 0x3F);
        } else if (code > 0x80) {
            byteArray[0] = 0xC0 | ((code & 0x7C0) >>> 6);
            byteArray[1] = 0x80 | (code & 0x3F);
        } else {
            byteArray[0] = code;
        }

        this.parsedData.push(byteArray);
    }

    this.parsedData = Array.prototype.concat.apply([], this.parsedData);

    if (this.parsedData.length != this.data.length) {
        this.parsedData.unshift(191);
        this.parsedData.unshift(187);
        this.parsedData.unshift(239);
    }
}

QR8bitByte.prototype = {
    getLength: function (buffer) {
        return this.parsedData.length;
    },
    write: function (buffer) {
        for (var i = 0, l = this.parsedData.length; i < l; i++) {
            buffer.put(this.parsedData[i], 8);
        }
    }
};
```

```
    function QRCodeModel(typeNumber, errorCorrectLevel) {
        this.typeNumber = typeNumber;
        this.errorCorrectLevel = errorCorrectLevel;
        this.modules = null;
        this.moduleCount = 0;
        this.dataCache = null;
        this.dataList = [];
    }

    QRCodeModel.prototype={addData:function(data){var newData=new
QR8bitByte(data);this.dataList.push(newData);this.dataCache=null;},isDark:function(row,col
){if(row<0||this.moduleCount<=row||col<0||this.moduleCount<=col){throw new
Error(row+","+col);}
    return this.modules[row][col];},getModuleCount:function(){return
this.moduleCount;},make:function(){this.makeImpl(false,this.getBestMaskPattern());},makeIm
pl:function(test,maskPattern){this.moduleCount=this.typeNumber*4+17;this.modules=new
Array(this.moduleCount);for(var row=0;row<this.moduleCount;row++){this.modules[row]=new
Array(this.moduleCount);for(var
col=0;col<this.moduleCount;col++){this.modules[row][col]=null;}}
    this.setupPositionProbePattern(0,0);this.setupPositionProbePattern(this.moduleCount-
7,0);this.setupPositionProbePattern(0,this.moduleCount-
7);this.setupPositionAdjustPattern();this.setupTimingPattern();this.setupTypeInfo(test,mas
kPattern);if(this.typeNumber>=7){this.setupTypeNumber(test);}
    if(this.dataCache==null){this.dataCache=QRCodeModel.createData(this.typeNumber,this.er
rorCorrectLevel,this.dataList);}
    this.mapData(this.dataCache,maskPattern);},setupPositionProbePattern:function(row,col)
{for(var r=-1;r<=7;r++){if(row+r<=-1||this.moduleCount<=row+r)continue;for(var c=-
1;c<=7;c++){if(col+c<=-
1||this.moduleCount<=col+c)continue;if((0<=r&&r<=6&&(c==0||c==6))||(0<=c&&c<=6&&(r==0||r==
6))||(2<=r&&r<=4&&2<=c&&c<=4)){this.modules[row+r][col+c]=true;}else{this.modules[row+r][c
ol+c]=false;}}}},getBestMaskPattern:function(){var minLostPoint=0;var pattern=0;for(var
i=0;i<8;i++){this.makeImpl(true,i);var
lostPoint=QRUtil.getLostPoint(this);if(i==0||minLostPoint>lostPoint){minLostPoint=lostPoin
t;pattern=i;}}
    return pattern;},createMovieClip:function(target_mc,instance_name,depth){var
qr_mc=target_mc.createEmptyMovieClip(instance_name,depth);var cs=1;this.make();for(var
row=0;row<this.modules.length;row++){var y=row*cs;for(var
col=0;col<this.modules[row].length;col++){var x=col*cs;var
dark=this.modules[row][col];if(dark){qr_mc.beginFill(0,100);qr_mc.moveTo(x,y);qr_mc.lineTo
(x+cs,y);qr_mc.lineTo(x+cs,y+cs);qr_mc.lineTo(x,y+cs);qr_mc.endFill();}}}
    return qr_mc;},setupTimingPattern:function(){for(var r=8;r<this.moduleCount-
8;r++){if(this.modules[r][6]!=null){continue;}
    this.modules[r][6]=(r%2==0);}
    for(var c=8;c<this.moduleCount-8;c++){if(this.modules[6][c]!=null){continue;}
    this.modules[6][c]=(c%2==0);}},setupPositionAdjustPattern:function(){var
pos=QRUtil.getPatternPosition(this.typeNumber);for(var i=0;i<pos.length;i++){for(var
j=0;j<pos.length;j++){var row=pos[i];var
col=pos[j];if(this.modules[row][col]!=null){continue;}
    for(var r=-2;r<=2;r++){for(var c=-2;c<=2;c++){if(r==-2||r==2||c==-
2||c==2||(r==0&&c==0)){this.modules[row+r][col+c]=true;}else{this.modules[row+r][col+c]=fa
lse;}}}}},setupTypeNumber:function(test){var
```

```
bits=QRUtil.getBCHTypeNumber(this.typeNumber);for(var i=0;i<18;i++){var
mod=(!test&&((bits>>i)&1)==1);this.modules[Math.floor(i/3)][i%3+this.moduleCount-8-
3]=mod;}
    for(var i=0;i<18;i++){var
mod=(!test&&((bits>>i)&1)==1);this.modules[i%3+this.moduleCount-8-
3][Math.floor(i/3)]=mod;}},setupTypeInfo:function(test,maskPattern){var
data=(this.errorCorrectLevel<<3)|maskPattern;var bits=QRUtil.getBCHTypeInfo(data);for(var
i=0;i<15;i++){var mod=(!test&&((bits>>i)&1)==1);if(i<6){this.modules[i][8]=mod;}else
if(i<8){this.modules[i+1][8]=mod;}else{this.modules[this.moduleCount-15+i][8]=mod;}}
    for(var i=0;i<15;i++){var
mod=(!test&&((bits>>i)&1)==1);if(i<8){this.modules[8][this.moduleCount-i-1]=mod;}else
if(i<9){this.modules[8][15-i-1+1]=mod;}else{this.modules[8][15-i-1]=mod;}}
    this.modules[this.moduleCount-8][8]=(!test);},mapData:function(data,maskPattern){var
inc=-1;var row=this.moduleCount-1;var bitIndex=7;var byteIndex=0;for(var
col=this.moduleCount-1;col>0;col-=2){if(col==6)col--;while(true){for(var
c=0;c<2;c++){if(this.modules[row][col-c]==null){var
dark=false;if(byteIndex<data.length){dark=(((data[byteIndex]>>>bitIndex)&1)==1);}
    var mask=QRUtil.getMask(maskPattern,row,col-c);if(mask){dark=!dark;}
    this.modules[row][col-c]=dark;bitIndex--;if(bitIndex==-1){byteIndex++;bitIndex=7;}}}
    row+=inc;if(row<0||this.moduleCount<=row){row-=inc;inc=-
inc;break;}}}}};QRCodeModel.PAD0=0xEC;QRCodeModel.PAD1=0x11;QRCodeModel.createData=functio
n(typeNumber,errorCorrectLevel,dataList){var
rsBlocks=QRRSBlock.getRSBlocks(typeNumber,errorCorrectLevel);var buffer=new
QRBitBuffer();for(var i=0;i<dataList.length;i++){var
data=dataList[i];buffer.put(data.mode,4);buffer.put(data.getLength(),QRUtil.getLengthInBit
s(data.mode,typeNumber));data.write(buffer);}
    var totalDataCount=0;for(var
i=0;i<rsBlocks.length;i++){totalDataCount+=rsBlocks[i].dataCount;}
    if(buffer.getLengthInBits()>totalDataCount*8){throw new Error("code length overflow.
("
    +buffer.getLengthInBits()
    +">"
    +totalDataCount*8
    +")");}
    if(buffer.getLengthInBits()+4<=totalDataCount*8){buffer.put(0,4);}
    while(buffer.getLengthInBits()%8!=0){buffer.putBit(false);}
    while(true){if(buffer.getLengthInBits()>=totalDataCount*8){break;}
    buffer.put(QRCodeModel.PAD0,8);if(buffer.getLengthInBits()>=totalDataCount*8){break;}
    buffer.put(QRCodeModel.PAD1,8);}
    return
QRCodeModel.createBytes(buffer,rsBlocks);};QRCodeModel.createBytes=function(buffer,rsBlock
s){var offset=0;var maxDcCount=0;var maxEcCount=0;var dcdata=new
Array(rsBlocks.length);var ecdata=new Array(rsBlocks.length);for(var
r=0;r<rsBlocks.length;r++){var dcCount=rsBlocks[r].dataCount;var
ecCount=rsBlocks[r].totalCount-
dcCount;maxDcCount=Math.max(maxDcCount,dcCount);maxEcCount=Math.max(maxEcCount,ecCount);dc
data[r]=new Array(dcCount);for(var
i=0;i<dcdata[r].length;i++){dcdata[r][i]=0xff&buffer.buffer[i+offset];}
    offset+=dcCount;var rsPoly=QRUtil.getErrorCorrectPolynomial(ecCount);var rawPoly=new
QRPolynomial(dcdata[r],rsPoly.getLength()-1);var modPoly=rawPoly.mod(rsPoly);ecdata[r]=new
Array(rsPoly.getLength()-1);for(var i=0;i<ecdata[r].length;i++){var
```

```
modIndex=i+modPoly.getLength()-
ecdata[r].length;ecdata[r][i]=(modIndex>=0)?modPoly.get(modIndex):0;}}
    var totalCodeCount=0;for(var
i=0;i<rsBlocks.length;i++){totalCodeCount+=rsBlocks[i].totalCount;}
    var data=new Array(totalCodeCount);var index=0;for(var i=0;i<maxDcCount;i++){for(var
r=0;r<rsBlocks.length;r++){if(i<dcdata[r].length){data[index++]=dcdata[r][i];}}}
    for(var i=0;i<maxEcCount;i++){for(var
r=0;r<rsBlocks.length;r++){if(i<ecdata[r].length){data[index++]=ecdata[r][i];}}}
    return data;};var
QRMode={MODE_NUMBER:1<<0,MODE_ALPHA_NUM:1<<1,MODE_8BIT_BYTE:1<<2,MODE_KANJI:1<<3};var
QRErrorCorrectLevel={L:1,M:0,Q:3,H:2};var
QRMaskPattern={PATTERN000:0,PATTERN001:1,PATTERN010:2,PATTERN011:3,PATTERN100:4,PATTERN101
:5,PATTERN110:6,PATTERN111:7};var
QRUtil={PATTERN_POSITION_TABLE:[[],[6,18],[6,22],[6,26],[6,30],[6,34],[6,22,38],[6,24,42],
[6,26,46],[6,28,50],[6,30,54],[6,32,58],[6,34,62],[6,26,46,66],[6,26,48,70],[6,26,50,74],[
6,30,54,78],[6,30,56,82],[6,30,58,86],[6,34,62,90],[6,28,50,72,94],[6,26,50,74,98],[6,30,5
4,78,102],[6,28,54,80,106],[6,32,58,84,110],[6,30,58,86,114],[6,34,62,90,118],[6,26,50,74,
98,122],[6,30,54,78,102,126],[6,26,52,78,104,130],[6,30,56,82,108,134],[6,34,60,86,112,138
],[6,30,58,86,114,142],[6,34,62,90,118,146],[6,30,54,78,102,126,150],[6,24,50,76,102,128,1
54],[6,28,54,80,106,132,158],[6,32,58,84,110,136,162],[6,26,54,82,110,138,166],[6,30,58,86
,114,142,170]],G15:(1<<10)|(1<<8)|(1<<5)|(1<<4)|(1<<2)|(1<<1)|(1<<0),G18:(1<<12)|(1<<11)|(
1<<10)|(1<<9)|(1<<8)|(1<<5)|(1<<2)|(1<<0),G15_MASK:(1<<14)|(1<<12)|(1<<10)|(1<<4)|(1<<1),g
etBCHTypeInfo:function(data){var d=data<<10;while(QRUtil.getBCHDigit(d)-
QRUtil.getBCHDigit(QRUtil.G15)>=0){d^=(QRUtil.G15<<(QRUtil.getBCHDigit(d)-
QRUtil.getBCHDigit(QRUtil.G15)));}
    return((data<<10)|d)^QRUtil.G15_MASK;},getBCHTypeNumber:function(data){var
d=data<<12;while(QRUtil.getBCHDigit(d)-
QRUtil.getBCHDigit(QRUtil.G18)>=0){d^=(QRUtil.G18<<(QRUtil.getBCHDigit(d)-
QRUtil.getBCHDigit(QRUtil.G18)));}
    return(data<<12)|d;},getBCHDigit:function(data){var
digit=0;while(data!=0){digit++;data>>>=1;}
    return digit;},getPatternPosition:function(typeNumber){return
QRUtil.PATTERN_POSITION_TABLE[typeNumber-
1];},getMask:function(maskPattern,i,j){switch(maskPattern){case
QRMaskPattern.PATTERN000:return(i+j)%2==0;case QRMaskPattern.PATTERN001:return i%2==0;case
QRMaskPattern.PATTERN010:return j%3==0;case QRMaskPattern.PATTERN011:return(i+j)%3==0;case
QRMaskPattern.PATTERN100:return(Math.floor(i/2)+Math.floor(j/3))%2==0;case
QRMaskPattern.PATTERN101:return(i*j)%2+(i*j)%3==0;case
QRMaskPattern.PATTERN110:return((i*j)%2+(i*j)%3)%2==0;case
QRMaskPattern.PATTERN111:return((i*j)%3+(i+j)%2)%2==0;default:throw new Error("bad
maskPattern:"+maskPattern);}},getErrorCorrectPolynomial:function(errorCorrectLength){var
a=new QRPolynomial([1],0);for(var i=0;i<errorCorrectLength;i++){a=a.multiply(new
QRPolynomial([1,QRMath.gexp(i)],0));}
    return a;},getLengthInBits:function(mode,type){if(1<=type&&type<10){switch(mode){case
QRMode.MODE_NUMBER:return 10;case QRMode.MODE_ALPHA_NUM:return 9;case
QRMode.MODE_8BIT_BYTE:return 8;case QRMode.MODE_KANJI:return 8;default:throw new
Error("mode:"+mode);}}else if(type<27){switch(mode){case QRMode.MODE_NUMBER:return 12;case
QRMode.MODE_ALPHA_NUM:return 11;case QRMode.MODE_8BIT_BYTE:return 16;case
QRMode.MODE_KANJI:return 10;default:throw new Error("mode:"+mode);}}else
if(type<41){switch(mode){case QRMode.MODE_NUMBER:return 14;case
QRMode.MODE_ALPHA_NUM:return 13;case QRMode.MODE_8BIT_BYTE:return 16;case
QRMode.MODE_KANJI:return 12;default:throw new Error("mode:"+mode);}}else{throw new
```

```
Error("type:"+type);}},getLostPoint:function(qrCode){var
moduleCount=qrCode.getModuleCount();var lostPoint=0;for(var
row=0;row<moduleCount;row++){for(var col=0;col<moduleCount;col++){var sameCount=0;var
dark=qrCode.isDark(row,col);for(var r=-
1;r<=1;r++){if(row+r<0||moduleCount<=row+r){continue;}
    for(var c=-1;c<=1;c++){if(col+c<0||moduleCount<=col+c){continue;}
    if(r==0&&c==0){continue;}
    if(dark==qrCode.isDark(row+r,col+c)){sameCount++;}}}
    if(sameCount>5){lostPoint+=(3+sameCount-5);}}}
    for(var row=0;row<moduleCount-1;row++){for(var col=0;col<moduleCount-1;col++){var
count=0;if(qrCode.isDark(row,col))count++;if(qrCode.isDark(row+1,col))count++;if(qrCode.is
Dark(row,col+1))count++;if(qrCode.isDark(row+1,col+1))count++;if(count==0||count==4){lostP
oint+=3;}}}
    for(var row=0;row<moduleCount;row++){for(var col=0;col<moduleCount-
6;col++){if(qrCode.isDark(row,col)&&!qrCode.isDark(row,col+1)&&qrCode.isDark(row,col+2)&&q
rCode.isDark(row,col+3)&&qrCode.isDark(row,col+4)&&!qrCode.isDark(row,col+5)&&qrCode.isDar
k(row,col+6)){lostPoint+=40;}}}
    for(var col=0;col<moduleCount;col++){for(var row=0;row<moduleCount-
6;row++){if(qrCode.isDark(row,col)&&!qrCode.isDark(row+1,col)&&qrCode.isDark(row+2,col)&&q
rCode.isDark(row+3,col)&&qrCode.isDark(row+4,col)&&!qrCode.isDark(row+5,col)&&qrCode.isDar
k(row+6,col)){lostPoint+=40;}}}
    var darkCount=0;for(var col=0;col<moduleCount;col++){for(var
row=0;row<moduleCount;row++){if(qrCode.isDark(row,col)){darkCount++;}}}
    var ratio=Math.abs(100*darkCount/moduleCount/moduleCount-
50)/5;lostPoint+=ratio*10;return lostPoint;}};var QRMath={glog:function(n){if(n<1){throw
new Error("glog("+n+")");}
    return QRMath.LOG_TABLE[n];},gexp:function(n){while(n<0){n+=255;}
    while(n>=256){n-=255;}
    return QRMath.EXP_TABLE[n];},EXP_TABLE:new Array(256),LOG_TABLE:new
Array(256)};for(var i=0;i<8;i++){QRMath.EXP_TABLE[i]=1<<i;}
    for(var i=8;i<256;i++){QRMath.EXP_TABLE[i]=QRMath.EXP_TABLE[i-4]^QRMath.EXP_TABLE[i-
5]^QRMath.EXP_TABLE[i-6]^QRMath.EXP_TABLE[i-8];}
    for(var i=0;i<255;i++){QRMath.LOG_TABLE[QRMath.EXP_TABLE[i]]=i;}
    function QRPolynomial(num,shift){if(num.length==undefined){throw new
Error(num.length+"/"+shift);}
    var offset=0;while(offset<num.length&&num[offset]==0){offset++;}
    this.num=new Array(num.length-offset+shift);for(var i=0;i<num.length-
offset;i++){this.num[i]=num[i+offset];}}
    QRPolynomial.prototype={get:function(index){return
this.num[index];},getLength:function(){return this.num.length;},multiply:function(e){var
num=new Array(this.getLength()+e.getLength()-1);for(var
i=0;i<this.getLength();i++){for(var
j=0;j<e.getLength();j++){num[i+j]^=QRMath.gexp(QRMath.glog(this.get(i))+QRMath.glog(e.get(
j)));}}
    return new QRPolynomial(num,0);},mod:function(e){if(this.getLength()-
e.getLength()<0){return this;}
    var ratio=QRMath.glog(this.get(0))-QRMath.glog(e.get(0));var num=new
Array(this.getLength());for(var i=0;i<this.getLength();i++){num[i]=this.get(i);}
    for(var i=0;i<e.getLength();i++){num[i]^=QRMath.gexp(QRMath.glog(e.get(i))+ratio);}
    return new QRPolynomial(num,0).mod(e);}};function
QRRSBlock(totalCount,dataCount){this.totalCount=totalCount;this.dataCount=dataCount;}
```

```
    QRRSBlock.RS_BLOCK_TABLE=[[1,26,19],[1,26,16],[1,26,13],[1,26,9],[1,44,34],[1,44,28],[
1,44,22],[1,44,16],[1,70,55],[1,70,44],[2,35,17],[2,35,13],[1,100,80],[2,50,32],[2,50,24],
[4,25,9],[1,134,108],[2,67,43],[2,33,15,2,34,16],[2,33,11,2,34,12],[2,86,68],[4,43,27],[4,
43,19],[4,43,15],[2,98,78],[4,49,31],[2,32,14,4,33,15],[4,39,13,1,40,14],[2,121,97],[2,60,
38,2,61,39],[4,40,18,2,41,19],[4,40,14,2,41,15],[2,146,116],[3,58,36,2,59,37],[4,36,16,4,3
7,17],[4,36,12,4,37,13],[2,86,68,2,87,69],[4,69,43,1,70,44],[6,43,19,2,44,20],[6,43,15,2,4
4,16],[4,101,81],[1,80,50,4,81,51],[4,50,22,4,51,23],[3,36,12,8,37,13],[2,116,92,2,117,93]
,[6,58,36,2,59,37],[4,46,20,6,47,21],[7,42,14,4,43,15],[4,133,107],[8,59,37,1,60,38],[8,44
,20,4,45,21],[12,33,11,4,34,12],[3,145,115,1,146,116],[4,64,40,5,65,41],[11,36,16,5,37,17]
,[11,36,12,5,37,13],[5,109,87,1,110,88],[5,65,41,5,66,42],[5,54,24,7,55,25],[11,36,12],[5,
122,98,1,123,99],[7,73,45,3,74,46],[15,43,19,2,44,20],[3,45,15,13,46,16],[1,135,107,5,136,
108],[10,74,46,1,75,47],[1,50,22,15,51,23],[2,42,14,17,43,15],[5,150,120,1,151,121],[9,69,
43,4,70,44],[17,50,22,1,51,23],[2,42,14,19,43,15],[3,141,113,4,142,114],[3,70,44,11,71,45]
,[17,47,21,4,48,22],[9,39,13,16,40,14],[3,135,107,5,136,108],[3,67,41,13,68,42],[15,54,24,
5,55,25],[15,43,15,10,44,16],[4,144,116,4,145,117],[17,68,42],[17,50,22,6,51,23],[19,46,16
,6,47,17],[2,139,111,7,140,112],[17,74,46],[7,54,24,16,55,25],[34,37,13],[4,151,121,5,152,
122],[4,75,47,14,76,48],[11,54,24,14,55,25],[16,45,15,14,46,16],[6,147,117,4,148,118],[6,7
3,45,14,74,46],[11,54,24,16,55,25],[30,46,16,2,47,17],[8,132,106,4,133,107],[8,75,47,13,76
,48],[7,54,24,22,55,25],[22,45,15,13,46,16],[10,142,114,2,143,115],[19,74,46,4,75,47],[28,
50,22,6,51,23],[33,46,16,4,47,17],[8,152,122,4,153,123],[22,73,45,3,74,46],[8,53,23,26,54,
24],[12,45,15,28,46,16],[3,147,117,10,148,118],[3,73,45,23,74,46],[4,54,24,31,55,25],[11,4
5,15,31,46,16],[7,146,116,7,147,117],[21,73,45,7,74,46],[1,53,23,37,54,24],[19,45,15,26,46
,16],[5,145,115,10,146,116],[19,75,47,10,76,48],[15,54,24,25,55,25],[23,45,15,25,46,16],[1
3,145,115,3,146,116],[2,74,46,29,75,47],[42,54,24,1,55,25],[23,45,15,28,46,16],[17,145,115
],[10,74,46,23,75,47],[10,54,24,35,55,25],[19,45,15,35,46,16],[17,145,115,1,146,116],[14,7
4,46,21,75,47],[29,54,24,19,55,25],[11,45,15,46,46,16],[13,145,115,6,146,116],[14,74,46,23
,75,47],[44,54,24,7,55,25],[59,46,16,1,47,17],[12,151,121,7,152,122],[12,75,47,26,76,48],[
39,54,24,14,55,25],[22,45,15,41,46,16],[6,151,121,14,152,122],[6,75,47,34,76,48],[46,54,24
,10,55,25],[2,45,15,64,46,16],[17,152,122,4,153,123],[29,74,46,14,75,47],[49,54,24,10,55,2
5],[24,45,15,46,46,16],[4,152,122,18,153,123],[13,74,46,32,75,47],[48,54,24,14,55,25],[42,
45,15,32,46,16],[20,147,117,4,148,118],[40,75,47,7,76,48],[43,54,24,22,55,25],[10,45,15,67
,46,16],[19,148,118,6,149,119],[18,75,47,31,76,48],[34,54,24,34,55,25],[20,45,15,61,46,16]
];QRRSBlock.getRSBlocks=function(typeNumber,errorCorrectLevel){var
rsBlock=QRRSBlock.getRsBlockTable(typeNumber,errorCorrectLevel);if(rsBlock==undefined){thr
ow new Error("bad rs block @
typeNumber:"+typeNumber+"/errorCorrectLevel:"+errorCorrectLevel);}
    var length=rsBlock.length/3;var list=[];for(var i=0;i<length;i++){var
count=rsBlock[i*3+0];var totalCount=rsBlock[i*3+1];var dataCount=rsBlock[i*3+2];for(var
j=0;j<count;j++){list.push(new QRRSBlock(totalCount,dataCount));}}
    return
list;};QRRSBlock.getRsBlockTable=function(typeNumber,errorCorrectLevel){switch(errorCorrec
tLevel){case QRErrorCorrectLevel.L:return QRRSBlock.RS_BLOCK_TABLE[(typeNumber-
1)*4+0];case QRErrorCorrectLevel.M:return QRRSBlock.RS_BLOCK_TABLE[(typeNumber-
1)*4+1];case QRErrorCorrectLevel.Q:return QRRSBlock.RS_BLOCK_TABLE[(typeNumber-
1)*4+2];case QRErrorCorrectLevel.H:return QRRSBlock.RS_BLOCK_TABLE[(typeNumber-
1)*4+3];default:return undefined;}};function QRBitBuffer(){this.buffer=[];this.length=0;}
    QRBitBuffer.prototype={get:function(index){var
bufIndex=Math.floor(index/8);return((this.buffer[bufIndex]>>>(7-
index%8))&1)==1;},put:function(num,length){for(var
i=0;i<length;i++){this.putBit(((num>>>(length-i-
1))&1)==1);}},getLengthInBits:function(){return this.length;},putBit:function(bit){var
bufIndex=Math.floor(this.length/8);if(this.buffer.length<=bufIndex){this.buffer.push(0);}
```

```
    if(bit){this.buffer[bufIndex]|=(0x80>>>(this.length%8));}
    this.length++;}};var
QRCodeLimitLength=[[17,14,11,7],[32,26,20,14],[53,42,32,24],[78,62,46,34],[106,84,60,44],[
134,106,74,58],[154,122,86,64],[192,152,108,84],[230,180,130,98],[271,213,151,119],[321,25
1,177,137],[367,287,203,155],[425,331,241,177],[458,362,258,194],[520,412,292,220],[586,45
0,322,250],[644,504,364,280],[718,560,394,310],[792,624,442,338],[858,666,482,382],[929,71
1,509,403],[1003,779,565,439],[1091,857,611,461],[1171,911,661,511],[1273,997,715,535],[13
67,1059,751,593],[1465,1125,805,625],[1528,1190,868,658],[1628,1264,908,698],[1732,1370,98
2,742],[1840,1452,1030,790],[1952,1538,1112,842],[2068,1628,1168,898],[2188,1722,1228,958]
,[2303,1809,1283,983],[2431,1911,1351,1051],[2563,1989,1423,1093],[2699,2099,1499,1139],[2
809,2213,1579,1219],[2953,2331,1663,1273]];

    function _isSupportCanvas() {
        return typeof CanvasRenderingContext2D != "undefined";
    }

    // android 2.x doesn't support Data-URI spec
    function _getAndroid() {
        var android = false;
        var sAgent = navigator.userAgent;

        if (/android/i.test(sAgent)) { // android
            android = true;
            var aMat = sAgent.toString().match(/android ([0-9]\.[0-9])/i);

            if (aMat && aMat[1]) {
                android = parseFloat(aMat[1]);
            }
        }

        return android;
    }

    var svgDrawer = (function() {

        var Drawing = function (el, htOption) {
            this._el = el;
            this._htOption = htOption;
        };

        Drawing.prototype.draw = function (oQRCode) {
            var _htOption = this._htOption;
            var _el = this._el;
            var nCount = oQRCode.getModuleCount();
            var nWidth = Math.floor(_htOption.width / nCount);
            var nHeight = Math.floor(_htOption.height / nCount);

            this.clear();

            function makeSVG(tag, attrs) {
                var el = document.createElementNS('http://www.w3.org/2000/svg', tag);
                for (var k in attrs)
```

```
                    if (attrs.hasOwnProperty(k)) el.setAttribute(k, attrs[k]);
                return el;
            }

            var svg = makeSVG("svg" , {'viewBox': '0 0 ' + String(nCount) + " " +
String(nCount), 'width': '100%', 'height': '100%', 'fill': _htOption.colorLight});
            svg.setAttributeNS("http://www.w3.org/2000/xmlns/", "xmlns:xlink",
"http://www.w3.org/1999/xlink");
            _el.appendChild(svg);

            svg.appendChild(makeSVG("rect", {"fill": _htOption.colorLight, "width":
"100%", "height": "100%"}));
            svg.appendChild(makeSVG("rect", {"fill": _htOption.colorDark, "width": "1",
"height": "1", "id": "template"}));

            for (var row = 0; row < nCount; row++) {
                for (var col = 0; col < nCount; col++) {
                    if (oQRCode.isDark(row, col)) {
                        var child = makeSVG("use", {"x": String(col), "y": String(row)});
                        child.setAttributeNS("http://www.w3.org/1999/xlink", "href",
"#template")
                        svg.appendChild(child);
                    }
                }
            }
        };
        Drawing.prototype.clear = function () {
            while (this._el.hasChildNodes())
                this._el.removeChild(this._el.lastChild);
        };
        return Drawing;
    })();

    var useSVG = document.documentElement.tagName.toLowerCase() === "svg";

    // Drawing in DOM by using Table tag
    var Drawing = useSVG ? svgDrawer : !_isSupportCanvas() ? (function () {
        var Drawing = function (el, htOption) {
            this._el = el;
            this._htOption = htOption;
        };

        /**
         * Draw the QRCode
         *
         * @param {QRCode} oQRCode
         */
        Drawing.prototype.draw = function (oQRCode) {
            var _htOption = this._htOption;
            var _el = this._el;
            var nCount = oQRCode.getModuleCount();
            var nWidth = Math.floor(_htOption.width / nCount);
```

```
            var nHeight = Math.floor(_htOption.height / nCount);
            var aHTML = ['<table style="border:0;border-collapse:collapse;">'];

            for (var row = 0; row < nCount; row++) {
                aHTML.push('<tr>');

                for (var col = 0; col < nCount; col++) {
                    aHTML.push('<td style="border:0;border-
collapse:collapse;padding:0;margin:0;width:' + nWidth + 'px;height:' + nHeight +
'px;background-color:' + (oQRCode.isDark(row, col) ? _htOption.colorDark :
_htOption.colorLight) + ';"></td>');
                }

                aHTML.push('</tr>');
            }

            aHTML.push('</table>');
            _el.innerHTML = aHTML.join('');

            // Fix the margin values as real size.
            var elTable = _el.childNodes[0];
            var nLeftMarginTable = (_htOption.width - elTable.offsetWidth) / 2;
            var nTopMarginTable = (_htOption.height - elTable.offsetHeight) / 2;

            if (nLeftMarginTable > 0 && nTopMarginTable > 0) {
                elTable.style.margin = nTopMarginTable + "px " + nLeftMarginTable +
"px";
            }
        };

        /**
         * Clear the QRCode
         */
        Drawing.prototype.clear = function () {
            this._el.innerHTML = '';
        };

        return Drawing;
    })() : (function () { // Drawing in Canvas
        function _onMakeImage() {
            this._elImage.src = this._elCanvas.toDataURL("image/png");
            this._elImage.style.display = "block";
            this._elCanvas.style.display = "none";
        }

        // Android 2.1 bug workaround
        // http://code.google.com/p/android/issues/detail?id=5141
        if (this._android && this._android <= 2.1) {
            var factor = 1 / window.devicePixelRatio;
            var drawImage = CanvasRenderingContext2D.prototype.drawImage;
            CanvasRenderingContext2D.prototype.drawImage = function (image, sx, sy, sw,
sh, dx, dy, dw, dh) {
```

```
            if (("nodeName" in image) && /img/i.test(image.nodeName)) {
                for (var i = arguments.length - 1; i >= 1; i--) {
                    arguments[i] = arguments[i] * factor;
                }
            } else if (typeof dw == "undefined") {
                arguments[1] *= factor;
                arguments[2] *= factor;
                arguments[3] *= factor;
                arguments[4] *= factor;
            }

            drawImage.apply(this, arguments);
        };
    }

    /**
     * Check whether the user's browser supports Data URI or not
     *
     * @private
     * @param {Function} fSuccess Occurs if it supports Data URI
     * @param {Function} fFail Occurs if it doesn't support Data URI
     */
    function _safeSetDataURI(fSuccess, fFail) {
        var self = this;
        self._fFail = fFail;
        self._fSuccess = fSuccess;

        // Check it just once
        if (self._bSupportDataURI === null) {
            var el = document.createElement("img");
            var fOnError = function() {
                self._bSupportDataURI = false;

                if (self._fFail) {
                    self._fFail.call(self);
                }
            };
            var fOnSuccess = function() {
                self._bSupportDataURI = true;

                if (self._fSuccess) {
                    self._fSuccess.call(self);
                }
            };

            el.onabort = fOnError;
            el.onerror = fOnError;
            el.onload = fOnSuccess;
            el.src =
"
GIAXDIBKE0DHxgljNBAAO9TXL0Y4OHwAAAABJRU5ErkJggg==";  // the Image contains 1px data.
            return;
```

```
    } else if (self._bSupportDataURI === true && self._fSuccess) {
        self._fSuccess.call(self);
    } else if (self._bSupportDataURI === false && self._fFail) {
        self._fFail.call(self);
    }
};

/**
 * Drawing QRCode by using canvas
 *
 * @constructor
 * @param {HTMLElement} el
 * @param {Object} htOption QRCode Options
 */
var Drawing = function (el, htOption) {
    this._bIsPainted = false;
    this._android = _getAndroid();

    this._htOption = htOption;
    this._elCanvas = document.createElement("canvas");
    this._elCanvas.width = htOption.width;
    this._elCanvas.height = htOption.height;
    el.appendChild(this._elCanvas);
    this._el = el;
    this._oContext = this._elCanvas.getContext("2d");
    this._bIsPainted = false;
    this._elImage = document.createElement("img");
    this._elImage.alt = "Scan me!";
    this._elImage.style.display = "none";
    this._el.appendChild(this._elImage);
    this._bSupportDataURI = null;
};

/**
 * Draw the QRCode
 *
 * @param {QRCode} oQRCode
 */
Drawing.prototype.draw = function (oQRCode) {
    var _elImage = this._elImage;
    var _oContext = this._oContext;
    var _htOption = this._htOption;

    var nCount = oQRCode.getModuleCount();
    var nWidth = _htOption.width / nCount;
    var nHeight = _htOption.height / nCount;
    var nRoundedWidth = Math.round(nWidth);
    var nRoundedHeight = Math.round(nHeight);

    _elImage.style.display = "none";
    this.clear();
```

```
            for (var row = 0; row < nCount; row++) {
                for (var col = 0; col < nCount; col++) {
                    var bIsDark = oQRCode.isDark(row, col);
                    var nLeft = col * nWidth;
                    var nTop = row * nHeight;
                    _oContext.strokeStyle = bIsDark ? _htOption.colorDark :
_htOption.colorLight;
                    _oContext.lineWidth = 1;
                    _oContext.fillStyle = bIsDark ? _htOption.colorDark :
_htOption.colorLight;
                    _oContext.fillRect(nLeft, nTop, nWidth, nHeight);

                    // 안티 앨리어싱 방지 처리
                    _oContext.strokeRect(
                        Math.floor(nLeft) + 0.5,
                        Math.floor(nTop) + 0.5,
                        nRoundedWidth,
                        nRoundedHeight
                    );

                    _oContext.strokeRect(
                        Math.ceil(nLeft) - 0.5,
                        Math.ceil(nTop) - 0.5,
                        nRoundedWidth,
                        nRoundedHeight
                    );
                }
            }

            this._bIsPainted = true;
        };

        /**
         * Make the image from Canvas if the browser supports Data URI.
         */
        Drawing.prototype.makeImage = function () {
            if (this._bIsPainted) {
                _safeSetDataURI.call(this, _onMakeImage);
            }
        };

        /**
         * Return whether the QRCode is painted or not
         *
         * @return {Boolean}
         */
        Drawing.prototype.isPainted = function () {
            return this._bIsPainted;
        };

        /**
```

```
         * Clear the QRCode
         */
        Drawing.prototype.clear = function () {
            this._oContext.clearRect(0, 0, this._elCanvas.width, this._elCanvas.height);
            this._bIsPainted = false;
        };

        /**
         * @private
         * @param {Number} nNumber
         */
        Drawing.prototype.round = function (nNumber) {
            if (!nNumber) {
                return nNumber;
            }

            return Math.floor(nNumber * 1000) / 1000;
        };

        return Drawing;
    })();

    /**
     * Get the type by string length
     *
     * @private
     * @param {String} sText
     * @param {Number} nCorrectLevel
     * @return {Number} type
     */
    function _getTypeNumber(sText, nCorrectLevel) {
        var nType = 1;
        var length = _getUTF8Length(sText);

        for (var i = 0, len = QRCodeLimitLength.length; i <= len; i++) {
            var nLimit = 0;

            switch (nCorrectLevel) {
                case QRErrorCorrectLevel.L :
                    nLimit = QRCodeLimitLength[i][0];
                    break;
                case QRErrorCorrectLevel.M :
                    nLimit = QRCodeLimitLength[i][1];
                    break;
                case QRErrorCorrectLevel.Q :
                    nLimit = QRCodeLimitLength[i][2];
                    break;
                case QRErrorCorrectLevel.H :
                    nLimit = QRCodeLimitLength[i][3];
                    break;
            }
```

```
            if (length <= nLimit) {
                break;
            } else {
                nType++;
            }
        }

        if (nType > QRCodeLimitLength.length) {
            throw new Error("Too long data");
        }

        return nType;
    }

    function _getUTF8Length(sText) {
        var replacedText = encodeURI(sText).toString().replace(/\%[0-9a-fA-F]{2}/g, 'a');
        return replacedText.length + (replacedText.length != sText ? 3 : 0);
    }

    /**
     * @class QRCode
     * @constructor
     * @example
     * new QRCode(document.getElementById("test"), "http://jindo.dev.naver.com/collie");
     *
     * @example
     * var oQRCode = new QRCode("test", {
     *    text : "http://naver.com",
     *    width : 128,
     *    height : 128
     * });
     *
     * oQRCode.clear(); // Clear the QRCode.
     * oQRCode.makeCode("http://map.naver.com"); // Re-create the QRCode.
     *
     * @param {HTMLElement|String} el target element or 'id' attribute of element.
     * @param {Object|String} vOption
     * @param {String} vOption.text QRCode link data
     * @param {Number} [vOption.width=256]
     * @param {Number} [vOption.height=256]
     * @param {String} [vOption.colorDark="#000000"]
     * @param {String} [vOption.colorLight="#ffffff"]
     * @param {QRCode.CorrectLevel} [vOption.correctLevel=QRCode.CorrectLevel.H] [L|M|Q|H]
     */
    QRCode = function (el, vOption) {
        this._htOption = {
            width : 256,
            height : 256,
            typeNumber : 4,
            colorDark : "#000000",
            colorLight : "#ffffff",
            correctLevel : QRErrorCorrectLevel.H
```

```
        };

        if (typeof vOption === 'string') {
            vOption = {
                text : vOption
            };
        }

        // Overwrites options
        if (vOption) {
            for (var i in vOption) {
                this._htOption[i] = vOption[i];
            }
        }

        if (typeof el == "string") {
            el = document.getElementById(el);
        }

        if (this._htOption.useSVG) {
            Drawing = svgDrawer;
        }

        this._android = _getAndroid();
        this._el = el;
        this._oQRCode = null;
        this._oDrawing = new Drawing(this._el, this._htOption);

        if (this._htOption.text) {
            this.makeCode(this._htOption.text);
        }
    };

    /**
     * Make the QRCode
     *
     * @param {String} sText link data
     */
    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();
    };

    /**
     * Make the Image from Canvas element
     * - It occurs automatically
     * - Android below 3 doesn't support Data-URI spec.
```

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

    /**
     * Clear the QRCode
     */
    QRCode.prototype.clear = function () {
        this._oDrawing.clear();
    };

    /**
     * @name QRCode.CorrectLevel
     */
    QRCode.CorrectLevel = QRErrorCorrectLevel;
})();
```

1.1 Implementation

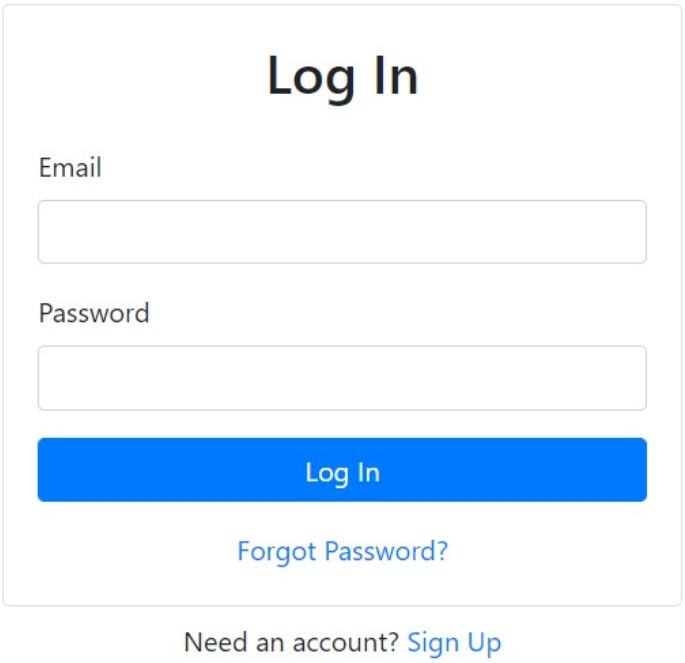


Fig1.1: Login with the registered credentials using email-id

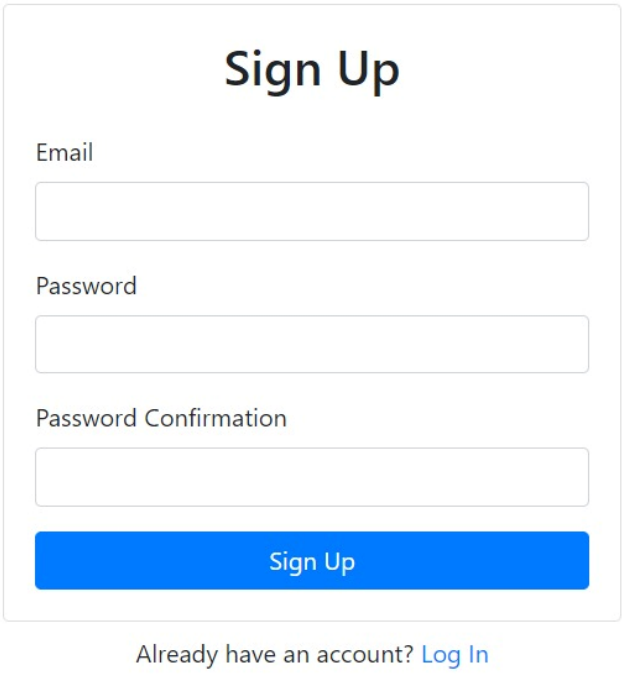


Fig 1.2: Use your credentials to signup

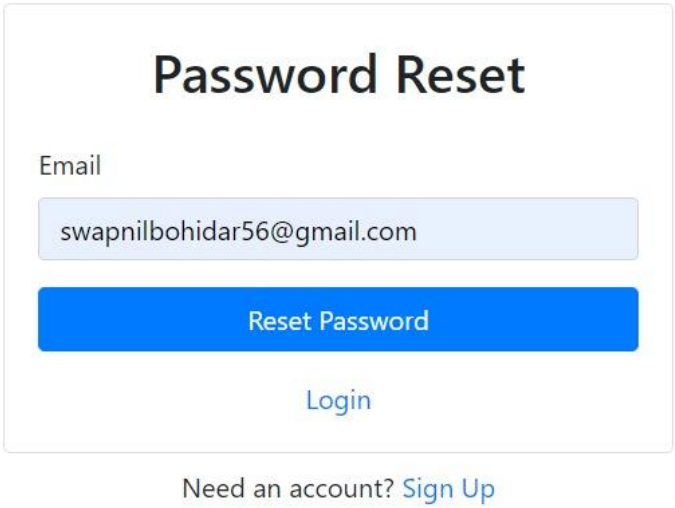
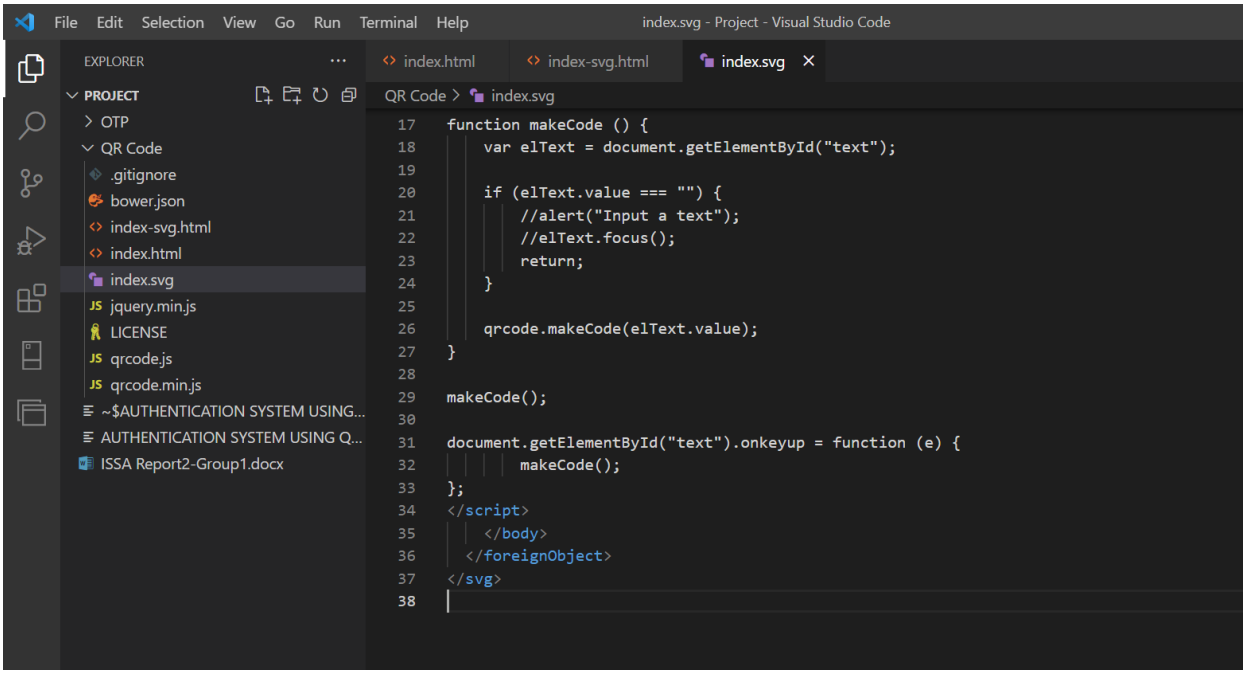


Fig 1.3 Password Reset Window

QR AUTHENTICATION



REFERENCES

[1] E. Barkan and E. Biham. Conditional Estimators: An Effective Attack on A5/1, pages 1–19. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[2] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In IEEE Symposium on Security and Privacy, pages 553–567. IEEE Computer Society, 2012.

[3] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. Technical Report 817, University of Cambridge Computer Laboratory, 2012.

[4] Y. Chow, W. Susilo, M. H. Au, and A. M. Barmawi. A visual one-time password authentication scheme using mobile devices. In L. C. K. Hui, S. H. Qing, E. Shi, and S. Yiu, editors, ICICS 2014, volume 8958 of Lecture Notes in Computer Science, pages 243–257. Springer, 2014.

[5] Y. Chow, W. Susilo, G. Yang, J. G. Phillips, I. Pranata, and A. M. Barmawi. Exploiting the error correction mechanism in QR codes for secret sharing. In J. K. Liu and R. Steinfeld, editors, ACISP 2016, volume 9722 of Lecture Notes in Computer Science, pages 1–17. Springer, 2016.

[6] D. S., Kim, B. H., & Lee, J. K., (2011) "A study on authentication system using QR code for mobile cloud computing environment", Future Information Technology, pp500-507.

[7] Kao, Y. W., Luo, G. H., Lin, H. T., Huang, Y. K., & Yuan, S. M., (2011) "Physical access control based on QR code" Cyber-enabled distributed computing and knowledge discovery pp285-288.

[8] H. Sun, "Cryptanalysis of password authentication schemes with smart cards," Information Security Conference 2001, pp. 221-223, May 2001.

[9] Financial security Agency, "Guide for End-to-End Encryption", 2007.10

[10] M.L. Das, A. Saxena, and V.P. Gulati, "A Dynamic ID-based Remote User Authentication Scheme," IEEE Transactions on Consumer Electronics, vol.50, no.2, pp. 629-631, May 2004.