COMPREHENSIVE PROJECT REPORT

ON

QR Code Generator & Information Extractor

An analytical study with reference to DELHI TECHNOLOGICAL UNIVERSITY SHAHBAD DAULATPUR ,DELHI

Submitted to

DELHI TECHNOLOGICAL UNIVERSITY

Under the guidance of

Dr. Seba Susan

Submitted by

Pulkit Chopra 2K19/IT/100

Shivam Garg 2K19/IT/117

B.TECH (2019-2023) Semester V

CANDIDATE'S DECLARATION

We (Pulkit Chopra & Shivam Garg), Roll No – 2K19/IT/100 & 2K19/IT/117, student of B.Tech. (INFORMATION TECHNOLOGY), hereby declare that the project Dissertation titled "QR Code Generator & Information Extractor" which is submitted by us to the Department of INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date: 11-11-2021

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

We hereby certify that the Project Dissertation titled “QR Code Generator & Information Extractor” which is submitted by Pulkit Chopra and Shivam Garg Roll No – 2K19/IT/100 & 2K19/IT/117; INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

DEPARTMENT OF INFORMATION TECHNOLOGY DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering) Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

We would like to convey our heartfelt thanks to our supervisor Dr Seba Susan for her ingenious ideas, tremendous help and cooperation.

We are extremely grateful to my friends who gave valuable suggestions and guidance for completion of my project. The cooperation and healthy criticism came handy and useful with them.

Finaly I would like to thank all the above mentioned people once again.

Abstract

Quick Response (QR) codes are most popular and can be found anywhere these days. We can spot many QR codes on textbooks, product ads, websites, product packaging and Nowadays payment apps are getting popular that uses QR code. The QR code can be used as one of the most fascinating ways of connecting consumers digitally via mobiles since the mobile phones have become a basic necessity thing of everyone & also to overcome the headache of sharing long paragraphs in the text format. So In this project, we present a prototype that creates QR codes by which the users enter text into a web browser and get the QR code generated and then another user can scan the generated QR code and the prototype will extract the useful information from the text using Reg-ex and auto fill-up the form.

Problem Description

In these new internet era everyone looking for a alternative to pass the Information efficiently. We took a reallife problem

1.Here a person wants to share a information with another person but doesn't want any third party apps or person to read or steal that information and also want to convey his message in a short form.

2.Here the receiver doesn't want to read the complete text of a sender as it can be a laborious task, He wants a program that can read and extract only that information that is useful to him and fill these up in the form (these receiver can be a merchant receiving heavy no. of registrations or anything)

PROJECT APPROACH

PROJECT ALGORITHM

We have chosen react JS for this project , React is a library of javascript for building user interfaces. Our code has been divided into 3 Parts:

1. Generate Your Own QR code.

2. Scan and Extract Information from the QR code.

3. Application like AUTO FILL FORM.

1 Generate:

To make information sharing easier and with high security we have made this app using a react component npm install qrcode.react

To make it user friendly we have set this component properties as size={290} and level={“H”} . Also we have added download functionality in our code. so that user can save the QR code (which has been generated) in the form of png.

To make the UI and make the user experience better we have implemented css styling in our code like adding background color, background image and with proper padding etc

Code:

Generator.js

```
import React,{useState} from 'react';
import './App.css'
var QRCode = require('qrcode.react');

const Generate = ()=>{
  const [value,setValue]=useState('');

  const handleChange = event=>{
    event.preventDefault();
    setValue(event.target.value)
  }

  const downloadQR = () => {
    const canvas = document.getElementById("123456");
    const pngUrl = canvas
      .toDataURL("image/png")
      .replace("image/png", "image/octet-stream");
    let downloadLink = document.createElement("a");
    downloadLink.href = pngUrl;
    downloadLink.download = "123456.png";
    document.body.appendChild(downloadLink);
    downloadLink.click();
    document.body.removeChild(downloadLink);
  };

  return(
```

```
    <div className="x">

  <div className="jumbotron bg-dark text-white">
  <h1 className="display-4">GENERATE YOUR QR CODE HERE</h1>

        <div className="form-group">
      <form>
        <label className="text-white text-center p-4">
          <h3>Enter The Info</h3>
          <textarea className="form-control"
style={{padding:"50px"}}  type="text" value={value} onChange={handleChange} />
        </label>
      </form>
      <QRCode
        id="123456"
        value={value}
        size={290}
        level={"H"}
        includeMargin={true}
      />
      <br></br>
      <a onClick={downloadQR}type="button" className="btn btn-success text-
white btn-lg p-2 my-2"> Download QR </a>
    </div>
    </div>
    </div>
  )
}
export default Generate;
```

App.css

```
.body {
  background-image: url("./img2.jpg");
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  height: 570px;
  overflow: hidden;
}
.x {
  background-color: darkslategray;
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  height: 100vh;
```

```
  overflow: hidden;
  width: 100%;
  text-align: center;
}
.z {
  overflow-x: hidden;
  overflow-y: hidden;
  background-color: darkslategray;
}
.head {
  background-color: black;
}
.jumbotron {
  background-color: darkgray;
}
```

2 Scan:

To scan the QR code we have used a react component , by using npm install --save react-qr-scanner

We have used properties like:

1 Delay - the delay between two scans. It should be initialised with a value in milliseconds and it has a default value of 500 milliseconds.

2 Facing Mode - We can choose which camera mode should be used Front or rear.

3 QRScan - It is a required function which gives the resulting value. event handler called after every scan and gives the result or null if no qr code is scanned.

The function of this component is to scan the QR Code and give us the resulting information ,all things are managed by the react hook (setState) . Initially we set the result value as an empty string , after every scan this react hook comes in the picture and sets the value of result equal to the required value.

Code

```
import React, { Component } from 'react'
import QrReader from 'react-qr-reader'
import extractEmails from './emailAddress'
import GetMobileNumber from './MobileNumber'
import getUrl from './url'
class Scan extends Component {
  state = {
    result: ''
  }

  handleScan = data => {
    if (data) {
      this.setState({
        result: data
      })
    }
  }
  handleError = err => {
    console.error(err)
  }

  render() {
    return (
      <div className="x">
      <div className=" row  text-light">
        <div className="col-6">
        <QrReader
          delay={300}
          onError={this.handleError}
          onScan={this.handleScan}
          style={{ width: '100%' }}

        />
         </div>
         {!this.state.result ? <h1 className="text-light">Welcome to Qr code
Info Extractor</h1> :
         <div className="col-6"
style={{textAlign:"start",fontWeight:"bold",fontSize:"25px"}}>
           <h1 className="text-light">Welcome to Qr code Info Extractor</h1>
           <p> Qr Code Conatins :  {this.state.result}</p>
           <br></br>
           <br></br>
        <p>Phone Number is {GetMobileNumber(this.state.result)}</p>
        <p>Email is {extractEmails(this.state.result)}</p>
        <p>Url is {getUrl(this.state.result)}</p>
```

```
      </div>}
      </div>
      </div>

    )
  }
}
export default Scan;
```

Then after we have implemented regex on this result

1. To extract the Phone number we have used text match(/([+]?\d{1,2}?)?(\d{3}?){2}\d{5}/g,' ') And then store all the phone numbers (present in the scanned result) in a map and then print them in the form of an unordered list.

   Code

```
const GetMobileNumber=(text)=>{
    var x =(text.match(/([+]?\d{1,2}?)?(\d{3}?){2}\d{5}/g,''))
    return !x ? null : (
      <div>
        <ul>
          {x.map(item=>{
            return <li>{item}</li>
          })}
        </ul>
    </div>
    )
  }
  export default GetMobileNumber;
```

2. To extract the Email address we have used text.match(/([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9._-]+)/gi,' ') And then store all the email address (present in the scanned result) in a map and then print them in the form of an unordered list.

Code

```
const extractEmails = (text)=>
{
    var x= text.match(/([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9._-
]+)/gi,'');
    return !x ? null : (
        <div>
          <ul>
            {x.map(item=>{
              return <li>{item}</li>
            })}
          </ul>
      </div>
      )
}
export default extractEmails
```

3. To extract the URL we have used text.match(/(https?:\/\/[^\s]+)/g,' ') And then store all the urls(present in the scanned result) in a map and then print them in the form of an unordered list.

Code

```
const getUrl=(text)=>{
    var Url =(text.match(/(https?:\/\/[^\s]+)/g))
    return !Url ? null : (
      <div>
        <ul>
          {
            Url.map(item =>{
              return (<li>
                  <a href={item}>{item}</a>
              </li>   )
            })
          }
        </ul>
      </div>
    )
  }
  export default getUrl;
```

3(SIGNUP FORM)

Using react and its hooks we have made an application to auto fill up the signup or any kind of form using regex

1 We have to scan a QR code

2 Retrieve the general things like email address, phone number from the scanned QR code with the help of regex.

3 Using react hooks we have made a function to auto fill the form after retrieval of details.

Code

```
import React,{Component, useState} from "react";
import QrReader from 'react-qr-reader'
import Se from "./S-Email";
import Sm from "./S-Mobile";
import './App.css'
class Email extends Component {
  state = {
    result: '',
    value : ''
  }

  handleScan = data => {
    if (data) {
      this.setState({
        result: data
      })
    }
  }
  handleError = err => {
    console.error(err)
  }


handleChangeE =event=>{
  console.log(event.target.value)
    this.setValue({value : event.target.value})

}
```

```
handleChangeM =event=>{
  console.log(event.target.value)
    this.setValue({value : event.target.value})

}
render(){
return (
  <div className="x">
  <div className=" row  text-white ">
        <div className="col-6">
        <QrReader
          delay={300}
          onError={this.handleError}
          onScan={this.handleScan}
          style={{ width: '100%' }}
        />
         </div>
  <div className="col-6">
    <h1 className="text-light">Welcome To SignUp Page</h1>
    <br></br>
<form>
  <div class="form-group">
    <label for="exampleInputEmail1"><h3>Email address</h3></label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-
describedby="emailHelp" value={Se(this.state.result)}
onChange={this.handleChangeE}/>
    <small id="emailHelp" class="form-text text-muted text-dark">We'll never
share your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputEmail1"><h3>Phone Number</h3></label>
    <input type="email" class="form-control"  value={Sm(this.state.result)}
onChange={this.handleChangeM}/>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1"><h3>Password</h3></label>
    <input type="password" class="form-control" id="exampleInputPassword1" />
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1" />
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-danger">Submit</button>
</form>
</div>
</div>
</div>
);
```

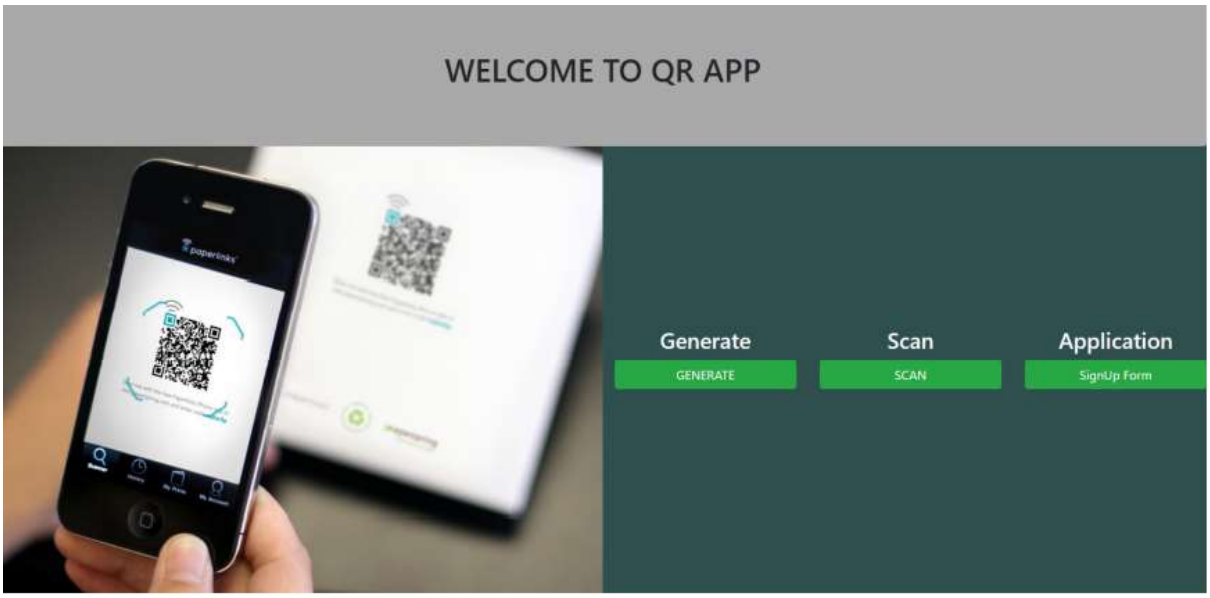
```
};
}

export default Email;
```

WEB DESIGN

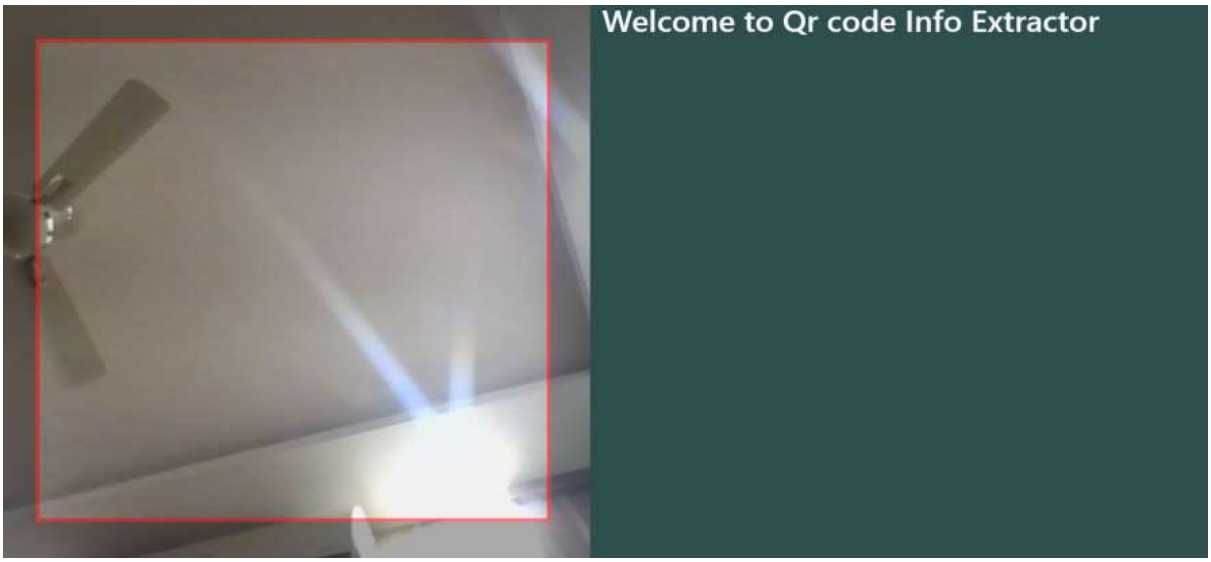
By using CSS , HTML with the react we have made a user friendly GUI which is easy to use and user can easily retrieve the data from the QR code and can also easily use it as an autofill signup form etc

HOMEPAGE

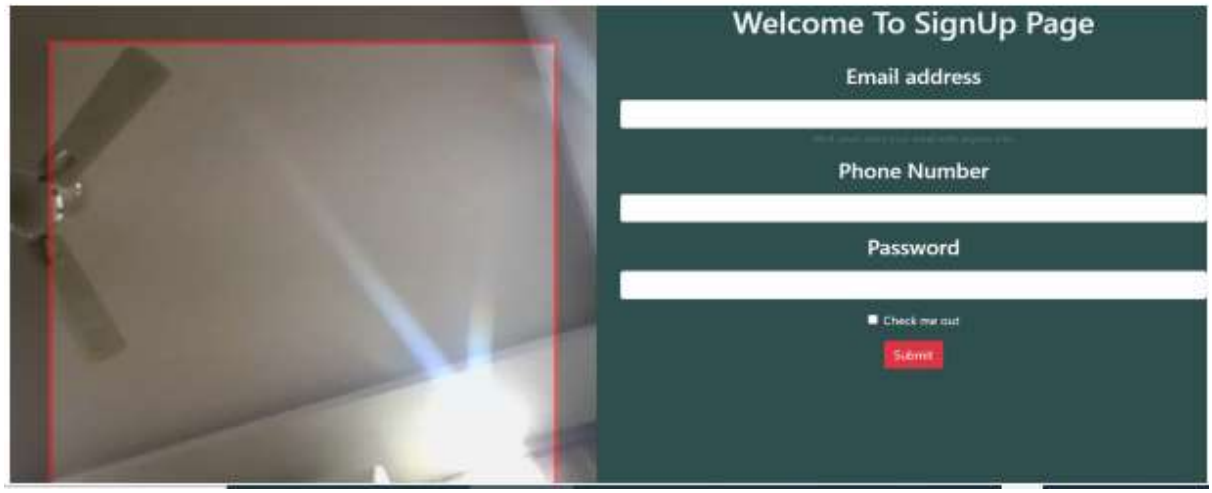


This is website's Homepage.

QR Code info Extractor



Code generator & Auto fill Form



INPUT FORMAT :

We have made a real time application in which

1. At generation of QR code user has to type the details

2. At the scanning section user has to display a Qr code in front of webcam

3. At the time if Autofill user has to show QR code in front of webcam.

OUTPUT FORMAT :

1. At generation section output will be downloaded in the user's device.

2. At scanning section output will be shown at the screen.

3. At Autofill section signup form will be automatically filled.

Conclusion

We have successfully implemented a project that can generate the QR code that is a compact form of large information and by using this generated QR Code we can auto fill any type application form.

Increase regular expression skills

Gain more knowledge about react js

REFERENCES

[1] https://www.npmjs.com/package/qrcode.react

[2] https://www.npmjs.com/package/react-qr-scanner