|  |  |
| --- | --- |
| **SIGCE.jpg** | JNIESTRT’S  **SMT. INDIRA GANDHI COLLEGE OF ENGINEERING**  GHANSOLI, NAVI MUMBAI – 400701  (Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)  **COMPUTER ENGINEERING DEPARTMENT**  **ACADEMIC YEAR :- 2019-20 (ODD SEM)** |

**Student Name:** ADITYA MAHIMKAR (24)

PRATIKSHA PATIL (40)

SIDDHI MHADLEKAR (27)

**Class:** SE **Sem:** III

**Subject:** OOPM **Course Outcome Covered:** CSL303.6

**MINI PROJECT**

TITLE: QR CODE GENERATOR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date of Performance** | **Date of Submission** | **Marks (05)** | | | **Sign / Remark** |
| **A** | **B** | **C** |
| **2** | **2** | **1** |
|  |  |  |  |  |  |
| **Total Marks** | | |
|  | | |

**A:** Understanding **B:** Implementation **C:** Oral

**Problem Statement**: Program to generate a QR Code (Quick Response Code) using Applets in Java.

**Implementation**: A basic applet is created with labels, textfields and buttons. When a string, especially a link, is being entered in the textfield, it is stored into a variable. The variables is been passed to the QR code library. It is then converted from string to Image type like .jpg, .png, etc. Its then passed to ByteArrayOutputStream in order to store in a file. The image is converted to byte array to store as FileOutputStream cannot pass byte data. A location has declared, here desktop, where the image file is been stored. On scanning the image using a QR Scanner App, the link is been displayed on the mobile.

JAR Files: Java Archive File.

This .jar files has been imported into program which generates the QR Code.

**Conclusion**: Creating this project, we learned the concepts of Applet and its implementation.

**References**: Most referred book was Programming with JAVA by Balaguruswamy. Further we referred to various sites over the internet for getting depth knowledge of applets.

**Program:**

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

import java.io.ByteArrayOutputStream;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import javax.swing.JOptionPane;

import net.glxn.qrgen.QRCode;

import net.glxn.qrgen.image.ImageType;

public class QRC extends Applet implements ActionListener {

Label l1 = new Label("Welcome to QR Code Generator", Label.CENTER);

Label l2 = new Label("Enter link or text:");

Label l3 = new Label("QR Code generated successfully on your Desktop.");

Button b = new Button("Generate");

TextField text = new TextField(50);

Font myFont1 = new Font("Times New Roman", Font.BOLD | Font.ITALIC, 22);

Font myFont2 = new Font("Times New Roman", Font.PLAIN, 17);

Font myFont3 = new Font("Times New Roman", Font.PLAIN, 14);

@Override

public void init() {

add(l1);

l1.setFont(myFont1);

add(l2);

l2.setFont(myFont2);

add(l3);

l3.setFont(myFont3);

add(text);

add(b);

b.addActionListener(this); }

@Override

public void actionPerformed(ActionEvent ae) {

try {new WriteText();

} catch (IOException e) {

e.printStackTrace();

}

}

public class WriteText {

WriteText() throws IOException {

String data = text.getText();

if (data.equals("")) {

JOptionPane.showMessageDialog(null, "Enter proper infomation!");

text.requestFocus();

} else {

ByteArrayOutputStream b = QRCode.from(data).to(ImageType.JPG).stream();

File fp = new File("C:\\Users\\Dell\\Desktop\\QRCode.jpg");

FileOutputStream fos = new FileOutputStream(fp);

fos.write(b.toByteArray());

fos.flush();

} } }

@Override

public void paint(Graphics g) {

Color color = **new** Color(250, 223, 173);

setBackground(color);

l1.setLocation(520, 14);

l2.setLocation(520, 60);

l3.setLocation(520, 560);

text.setLocation(520, 110);

b.setLocation(670, 160);

b.setSize(80, 50);

}

}

