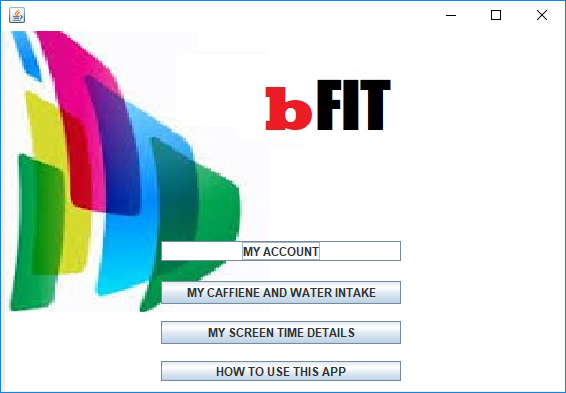
**D.A.V PUBLIC SCHOOL,**

**AUNDH, PUNE**



***IP Project***

***2019-20***

**bFIT Health Management Desktop Application**

**Submitted by - Vedang D. Asgaonkar**

**Class- XII –B**

**Roll No- 20**

**INDEX**

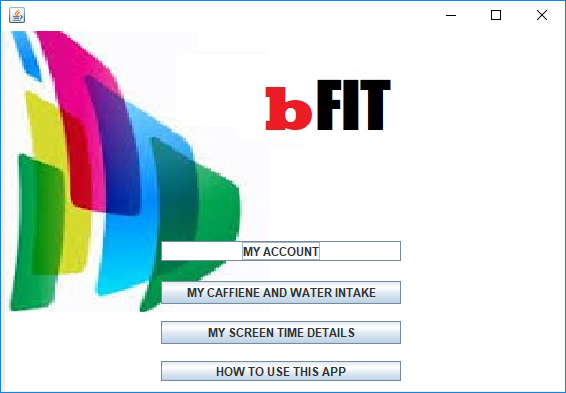
|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Details** | **Page No.** |
| 1. | OBJECTIVE | 3 |
| 2. | DESCRIPTION | 4 |
| 3. | JAVA CODE | 9 |
| 4. | MYSQL DATABASE | 36 |

**Objective**

Objective of this project is to build a desk top application that can be helpful for managing the health of desktop users. It mainly focuses on following benefits –

* Keeping measure of water intake, so that the user remains hydrated
* Keeping count of the caffeine intake from various drinks, so that user is reminded to keep control.
* Giving statistics of daily intake of water and coffee for easy visualization in a chart
* Advice user to shut down the screen or stop using screen for some time after a stipulated period, to avoid fatigue.
* Health advices every 4 hours

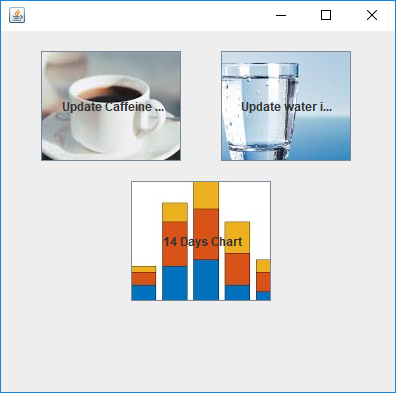
**Description**



Main screen of the application provides 2 options –

1. Go to Caffiene intake control and Water intake functionality screen
2. Go to a Screen Time warning functionality screen

**Caffiene and Water Intake Options**

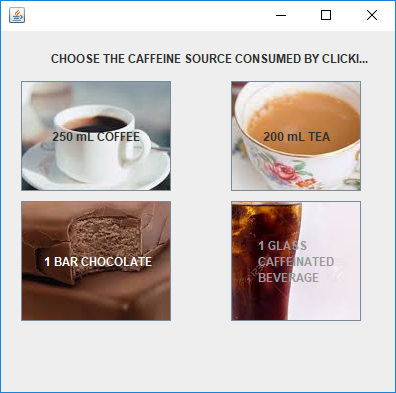


On this screen you get 3 options –

1. Go to Updating Caffiene intake
2. Go to Updating water intake
3. Go to Caffien and Water intake - Charts

**Caffiene Intake Counter**

****



This screen has been provided to update the intake of caffeine by the user during the day. There are 4 options for updating intake instances, in the form of –

1. Cup of coffee (~250 ml)
2. Cup of Tea (~200 ml)
3. 1 bar of chocolate
4. 1 glass caffeinated beverage

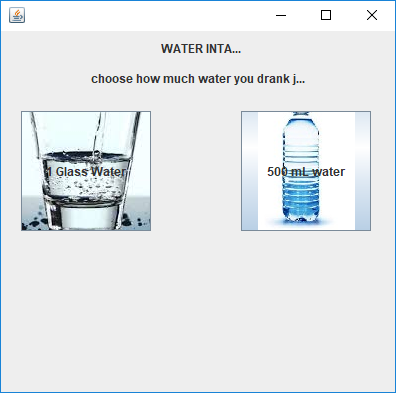
These statistics are stored in the database daily along with the day and time of consumption. Daily a new set of records are maintained.

**Warning functionality:**

An internal variable is used to sum up the amount of caffeine consumption. If the user has consumed caffeine above a particular threshold, a warning screen is popped up that advices the user to control further consumption.

**Water Intake Counter**

****



This screen has been provided to update the intake of water by the user during the day. There are 2 options –

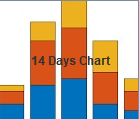
1. 1 Glass of water
2. 500 ml of water

These statistics are stored in the database along with the day and time of consumption. Daily a new set of records are maintained.

**Warning functionality:**

An internal counter is maintained which measures the time between two instances of water consumption. If the user has not consumed water for two hours, a warning is popped up on the screen to advice the .user to consume water.

**Caffeine Intake Statistics**

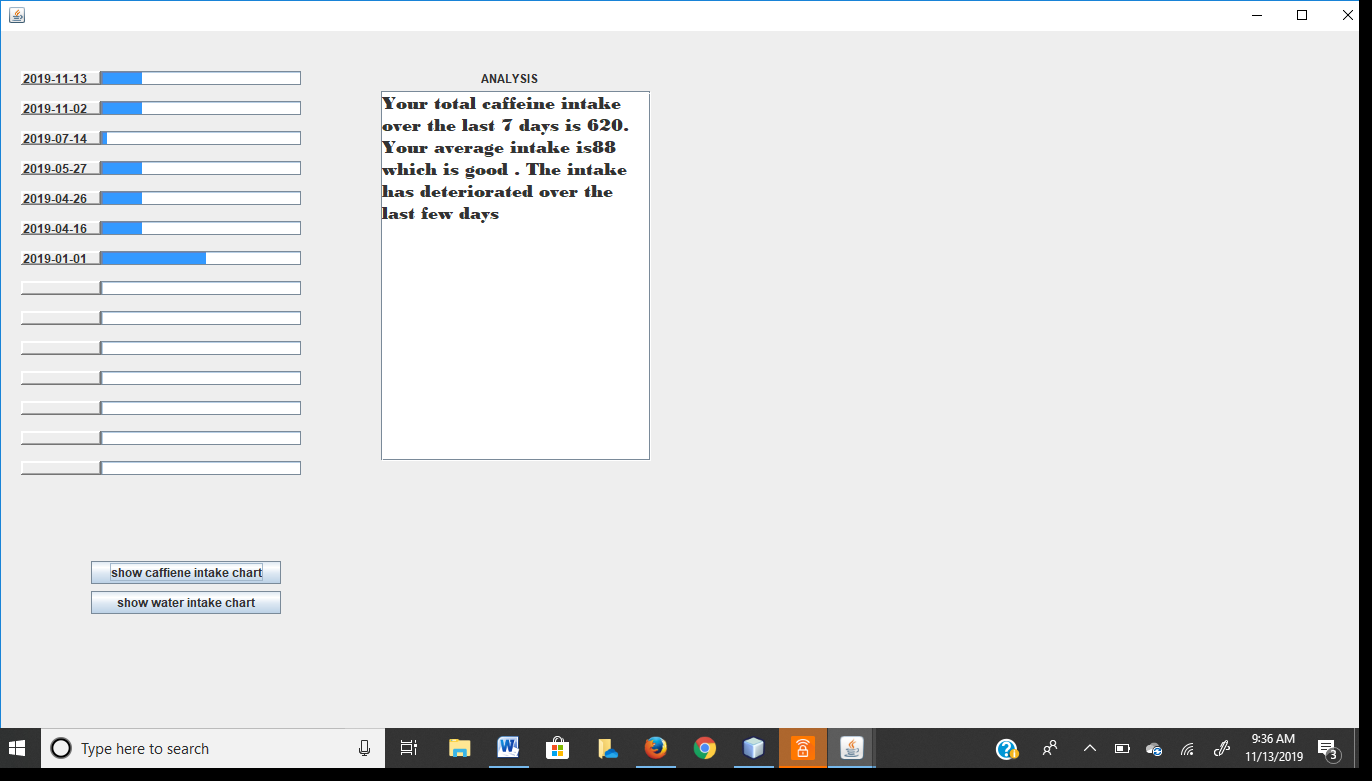


Here the user is shown a statistics screen, where he can see the with two options –

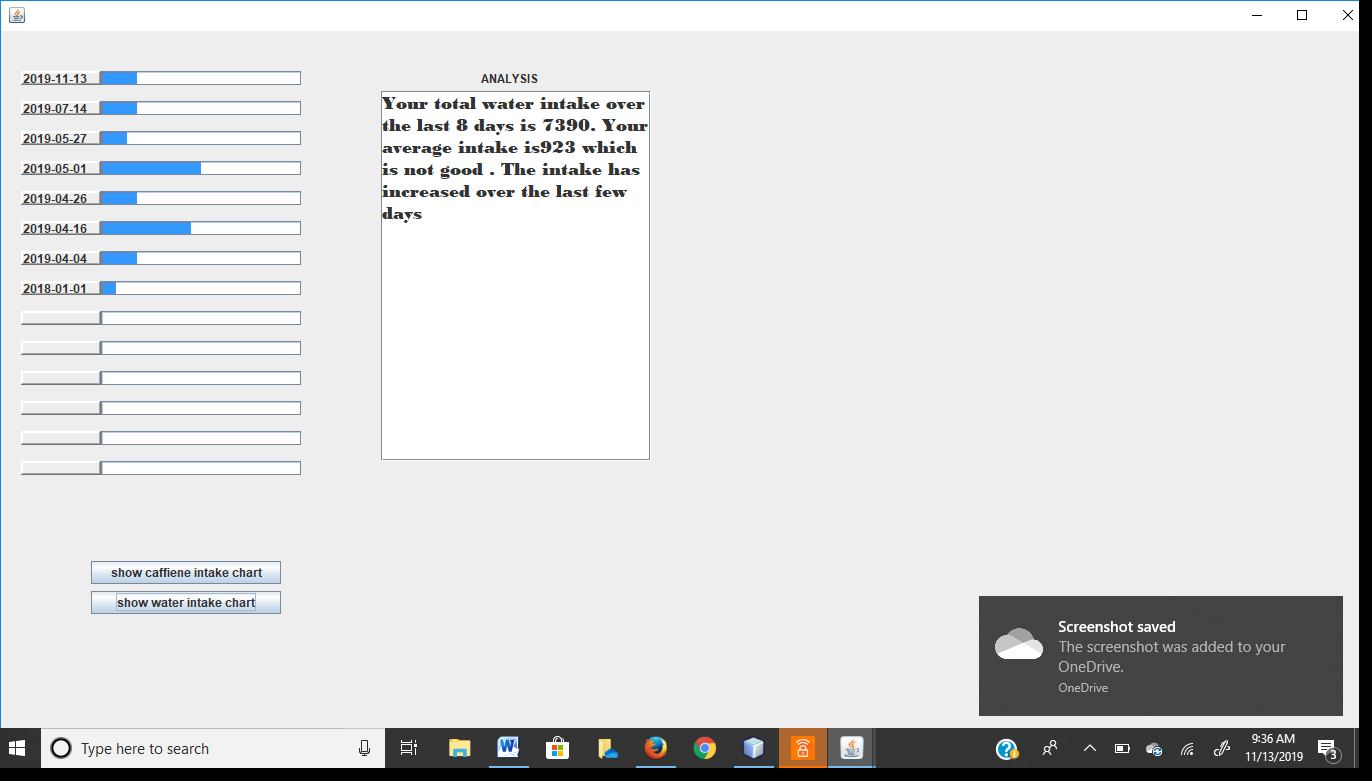
1. See statistics of water consumption over last 14 days
2. See statistics of coffee consumption over last 14 days

The consumption statistics are shown in the form of a graph along with Analysis in a dialog box.

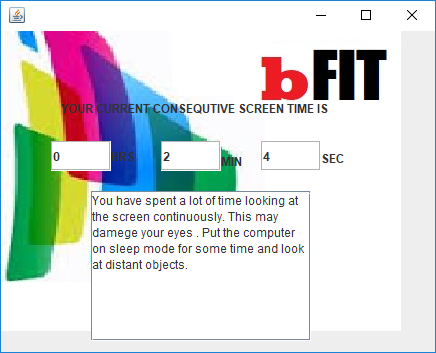
Caffiene intake chart and analysis -



Water intake chart and analysis -



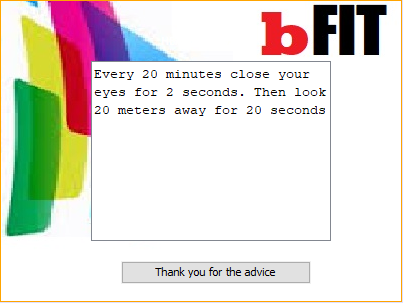
**Screen Usage Advice**



This dialog box pops up if the user has used the screen for a long time.

**Adhoc Health Advice**

4 hours after the screen starts and every 4 hours after that the software picks up a random advice among a set of 14 advices and shows the advice to the user in a pop-up screen.

.

**Java Program Code**

import javax.swing.Timer;

import java.awt.event.\*;

import java.util.Random;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

import java.sql.\*;

import java.awt.Dimension;

public class NewJFrame extends javax.swing.JFrame {

/\*\* Creates new form NewJFrame \*/

public NewJFrame() {

initComponents();

adviseTimer.start();

waterTimer.start();

screenTimer.start();

}

long screenTime = 0;

Timer adviseTimer = new Timer(14400000, new ActionListener() {

public void actionPerformed(ActionEvent evt) {

Random advisor = new Random();

int adv = advisor.nextInt(10);

System.out.println(adv);

adviceDialog.setVisible(true);

// creates a variable to produce one among 10 advices at random

switch (adv) {

case 0:

adviceTA.setText("Sitting for long periodds of time in front of a computer at weird angle or using keyboard inappropraiately can lead to issues. Sit in the chair squarly with your arms resting at 90 degrees while they type. Donot lean or slouch");

break;

case 1:

adviceTA.setText("Every 20 minutes look 20 metres away for 20 seconds ");

break;

case 2:

adviceTA.setText("Walk around in your room to reduce stiffness and relieve eyestrain. ");

break;

case 3:

adviceTA.setText("Simple stretching exercises can help relieve stiffness. Stand at you place and stretch you hands apart for 5-7 seconds.");

break;

case 4:

adviceTA.setText("listening to soothing music and meditating will relieve stress and help you start working with renewed energy.");

break;

case 5:

adviceTA.setText("If you have access to a treadmill jump onto it every morning before work.");

break;

case 6:

adviceTA.setText("Sit up straight. bad posturecan hurt your back and the pain can spread to your neck and head");

break;

case 7:

adviceTA.setText("Your monitor should be at eye level so that yo are not looking down constantly. The display should be at an arms length from your head.");

break;

case 8:

adviceTA.setText("A good office chair will give you some great support and will mitigate lower back pain.");

break;

case 9:

adviceTA.setText("Increase the size of your screen font so that your eyes dont strain much");

break;

}

adviceDialog.setVisible(true);

adviceDialog.requestFocus();

adviceDialog.setAlwaysOnTop(true);

}

});

Timer waterTimer = new Timer(3600000, new ActionListener() {

public void actionPerformed(ActionEvent evt) {

hydrateDialog.setVisible(true);

hydrateDialog.requestFocus();

}

});

Timer screenTimer = new Timer(1000, new ActionListener() {

public void actionPerformed(ActionEvent evt) {

screenTime++;

int hrs = (int) (screenTime / 3600);

int min = (int) ((screenTime - hrs \* 3600) / 60);

int sec = (int) (screenTime - ((hrs \* 3600) + (min \* 60)));

hrsLabel.setText("" + hrs);

minLabel.setText("" + min);

secLabel.setText("" + sec);

}

});

private void initComponents() {

private void formWindowActivated(java.awt.event.WindowEvent evt) {

this.setLocation(500, 0);

caffWaterFrame.setVisible(false);

// posture warning

}

private void walkThanksActionPerformed(java.awt.event.ActionEvent evt) {

adviceDialog.setVisible(false);

// TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

caffWaterFrame.setVisible(true);

// TODO add your handling code here:

}

private void caffIntakeBtnActionPerformed(java.awt.event.ActionEvent evt) {

caffeineFrame.setVisible(true);

// TODO add your handling code here:

}

private void waterIntakeBtnActionPerformed(java.awt.event.ActionEvent evt) {

waterFrame.setVisible(true);

// TODO add your handling code here:

}

private void glassWaterBtnActionPerformed(java.awt.event.ActionEvent evt) {

waterTimer.start();

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String q1 = "select \* from waterIntake where rdate = curdate();";

ResultSet rs = stmt.executeQuery(q1);

boolean rdateExists = false;

while (rs.next()) {

rdateExists = true;

}

if (rdateExists) {

String q2 = "update waterIntake set water= water+200 where rdate = curdate();";

stmt.executeUpdate(q2);

} else {

String query = "Insert into waterIntake values(curdate(),200);";

stmt.executeUpdate(query);

}

stmt.close();

con.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

// TODO add your handling code here:

}

private void bottleWaterBtnActionPerformed(java.awt.event.ActionEvent evt) {

waterTimer.start();

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String q1 = "select \* from waterIntake where rdate = curdate();";

ResultSet rs = stmt.executeQuery(q1);

boolean rdateExists = false;

while (rs.next()) {

rdateExists = true;

}

if (rdateExists) {

String q2 = "update waterIntake set water= water+500 where rdate = curdate();";

stmt.executeUpdate(q2);

} else {

String query = "Insert into waterIntake values(curdate(),500);";

stmt.executeUpdate(query);

}

stmt.close();

con.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

// TODO add your handling code here:

}

private void formWindowOpened(java.awt.event.WindowEvent evt) {

// TODO add your handling code here:

}

private void formWindowStateChanged(java.awt.event.WindowEvent evt) {

// TODO add your handling code here:

}

private void formWindowLostFocus(java.awt.event.WindowEvent evt) {

// TODO add your handling code here:

}

private void waterCharterBtnActionPerformed(java.awt.event.ActionEvent evt) {

p1.setValue(0);

p2.setValue(0);

p3.setValue(0);

p4.setValue(0);

p5.setValue(0);

p6.setValue(0);

p7.setValue(0);

p8.setValue(0);

p9.setValue(0);

p10.setValue(0);

p11.setValue(0);

p12.setValue(0);

p13.setValue(0);

p14.setValue(0);

analysisTA.setText("");

l1.setText("");

l2.setText("");

l3.setText("");

l4.setText("");

l5.setText("");

l6.setText("");

l7.setText("");

l8.setText("");

l9.setText("");

l10.setText("");

l11.setText("");

l12.setText("");

l13.setText("");

l14.setText("");

p1.setMaximum(4000);

p2.setMaximum(4000);

p3.setMaximum(4000);

p4.setMaximum(4000);

p5.setMaximum(4000);

p6.setMaximum(4000);

p7.setMaximum(4000);

p8.setMaximum(4000);

p9.setMaximum(4000);

p10.setMaximum(4000);

p11.setMaximum(4000);

p12.setMaximum(4000);

p13.setMaximum(4000);

p14.setMaximum(4000);

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String query = "Select distinct rdate,water from waterIntake order by rdate desc;";

ResultSet rs = stmt.executeQuery(query);

int i = 1;

int sum = 0;

int avg = 0;

int first = 0;

int last = 0;

while (rs.next() && i <= 14) {

int w = 0;

w = rs.getInt("water");

String date = rs.getString("rdate");

sum = sum + w;

avg = sum / i;

if (i == 1) {

first = w;

}

last = w;

switch (i) {

case 1:

p1.setValue(w);

l1.setText(date);

break;

case 2:

p2.setValue(w);

l2.setText(date);

break;

case 3:

p3.setValue(w);

l3.setText(date);

break;

case 4:

p4.setValue(w);

l4.setText(date);

break;

case 5:

p5.setValue(w);

l5.setText(date);

break;

case 6:

p6.setValue(w);

l6.setText(date);

break;

case 7:

p7.setValue(w);

l7.setText(date);

break;

case 8:

p8.setValue(w);

l8.setText(date);

break;

case 9:

p9.setValue(w);

l9.setText(date);

break;

case 10:

p10.setValue(w);

l10.setText(date);

break;

case 11:

p11.setValue(w);

l11.setText(date);

break;

case 12:

p12.setValue(w);

l12.setText(date);

break;

case 13:

p13.setValue(w);

l13.setText(date);

break;

case 14:

p14.setValue(w);

l14.setText(date);

break;

}

i++;

}

String intakeIsGoodOrNot = avg > 3700 ? "good" : "not good";

String IsIncreasing = (first - last > 0) ? "increased" : "deteriorated";

analysisTA.setText("Your total water intake over the last " + (i - 1) + " days is " + sum + ". Your average intake is" + avg + " which is " + intakeIsGoodOrNot + " . The intake has " + IsIncreasing + " over the last few days");

rs.close();

stmt.close();

con.close();

System.out.println(1);

} catch (Exception e) {

System.out.println(e.getMessage());

}

// TODO add your handling code here:

}

private void chartBtnActionPerformed(java.awt.event.ActionEvent evt) {

chartFrame.setVisible(true);

// TODO add your handling code here:

}

private void coffeeBtnActionPerformed(java.awt.event.ActionEvent evt) {

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String q1 = "select \* from caffeineIntake where rdate = curdate();";

ResultSet rs = stmt.executeQuery(q1);

boolean rdateExists = false;

while (rs.next()) {

rdateExists = true;

}

if (rdateExists) {

String q2 = "update caffeineIntake set caffiene= caffiene+80 where rdate = curdate();";

stmt.executeUpdate(q2);

} else {

String query = "Insert into caffeineIntake values(curdate(),80);";

stmt.executeUpdate(query);

}

stmt.close();

con.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

// TODO add your handling code here:

}

private void teaBtnActionPerformed(java.awt.event.ActionEvent evt) {

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String q1 = "select \* from caffeineIntake where rdate = curdate();";

ResultSet rs = stmt.executeQuery(q1);

boolean rdateExists = false;

while (rs.next()) {

rdateExists = true;

}

if (rdateExists) {

String q2 = "update caffeineIntake set caffiene= caffiene+30 where rdate = curdate();";

stmt.executeUpdate(q2);

} else {

String query = "Insert into caffeineIntake values(curdate(),30);";

stmt.executeUpdate(query);

}

stmt.close();

con.close();

} catch (Exception e) {

System.out.println(e.getMessage());

} // TODO add your handling code here:

}

private void chocBtnActionPerformed(java.awt.event.ActionEvent evt) {

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String q1 = "select \* from caffeineIntake where rdate = curdate();";

ResultSet rs = stmt.executeQuery(q1);

boolean rdateExists = false;

while (rs.next()) {

rdateExists = true;

}

if (rdateExists) {

String q2 = "update caffeineIntake set caffiene= caffiene+10 where rdate = curdate();";

stmt.executeUpdate(q2);

} else {

String query = "Insert into caffeineIntake values(curdate(),10);";

stmt.executeUpdate(query);

}

stmt.close();

con.close();

} catch (Exception e) {

System.out.println(e.getMessage());

} // TODO add your handling code here:

}

private void cokeBtnActionPerformed(java.awt.event.ActionEvent evt) {

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String q1 = "select \* from caffeineIntake where rdate = curdate();";

ResultSet rs = stmt.executeQuery(q1);

boolean rdateExists = false;

while (rs.next()) {

rdateExists = true;

}

if (rdateExists) {

String q2 = "update caffeineIntake set caffiene= caffiene+30 where rdate = curdate();";

stmt.executeUpdate(q2);

} else {

String query = "Insert into caffeineIntake values(curdate(),30);";

stmt.executeUpdate(query);

}

stmt.close();

con.close();

} catch (Exception e) {

System.out.println(e.getMessage());

}

// TODO add your handling code here:

}

private void caffeineCharterBtnActionPerformed(java.awt.event.ActionEvent evt) {

p1.setValue(0);

p2.setValue(0);

p3.setValue(0);

p4.setValue(0);

p5.setValue(0);

p6.setValue(0);

p7.setValue(0);

p8.setValue(0);

p9.setValue(0);

p10.setValue(0);

p11.setValue(0);

p12.setValue(0);

p13.setValue(0);

p14.setValue(0);

analysisTA.setText("");

l1.setText("");

l2.setText("");

l3.setText("");

l4.setText("");

l5.setText("");

l6.setText("");

l7.setText("");

l8.setText("");

l9.setText("");

l10.setText("");

l11.setText("");

l12.setText("");

l13.setText("");

l14.setText("");

p1.setMaximum(400);

p2.setMaximum(400);

p3.setMaximum(400);

p4.setMaximum(400);

p5.setMaximum(400);

p6.setMaximum(400);

p7.setMaximum(400);

p8.setMaximum(400);

p9.setMaximum(400);

p10.setMaximum(400);

p11.setMaximum(400);

p12.setMaximum(400);

p13.setMaximum(400);

p14.setMaximum(400);

try {

Class.forName("java.sql.DriverManager");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/fitapp", "root", "1234");

Statement stmt = con.createStatement();

String query = "Select distinct rdate,caffeine from caffeineIntake order by rdate desc;";

ResultSet rs = stmt.executeQuery(query);

int i = 1;

int sum = 0;

int avg = 0;

int first = 0;

int last = 0;

while (rs.next() && i <= 14) {

int w = 0;

w = rs.getInt("caffeine");

String date = rs.getString("rdate");

sum = sum + w;

avg = sum / i;

if (i == 1) {

first = w;

}

last = w;

switch (i) {

case 1:

p1.setValue(w);

l1.setText(date);

break;

case 2:

p2.setValue(w);

l2.setText(date);

break;

case 3:

p3.setValue(w);

l3.setText(date);

break;

case 4:

p4.setValue(w);

l4.setText(date);

break;

case 5:

p5.setValue(w);

l5.setText(date);

break;

case 6:

p6.setValue(w);

l6.setText(date);

break;

case 7:

p7.setValue(w);

l7.setText(date);

break;

case 8:

p8.setValue(w);

l8.setText(date);

break;

case 9:

p9.setValue(w);

l9.setText(date);

break;

case 10:

p10.setValue(w);

l10.setText(date);

break;

case 11:

p11.setValue(w);

l11.setText(date);

break;

case 12:

p12.setValue(w);

l12.setText(date);

break;

case 13:

p13.setValue(w);

l13.setText(date);

break;

case 14:

p14.setValue(w);

l14.setText(date);

break;

}

i++;

}

String intakeIsGoodOrNot = avg < 400 ? "good" : "not good";

String IsIncreasing = (first - last > 0) ? "increased" : "deteriorated";

analysisTA.setText("Your total caffeine intake over the last " + (i - 1) + " days is " + sum + ". Your average intake is" + avg + " which is " + intakeIsGoodOrNot + " . The intake has " + IsIncreasing + " over the last few days");

rs.close();

stmt.close();

con.close();

System.out.println(1);

} catch (Exception e) {

System.out.println(e.getMessage());

} // TODO add your handling code here:

}

private void screenTimeButtonActionPerformed(java.awt.event.ActionEvent evt) {

screenTimeFrame.setVisible(true);

// TODO add your handling code here:

}

private void screenTimeFrameWindowClosed(java.awt.event.WindowEvent evt) {

screenTimeAdviceTA.setVisible(false);

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new NewJFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JDialog adviceDialog;

private javax.swing.JTextArea adviceTA;

private javax.swing.JTextArea analysisTA;

private javax.swing.JButton bottleWaterBtn;

private javax.swing.JButton caffIntakeBtn;

private javax.swing.JFrame caffWaterFrame;

private javax.swing.JButton caffeineCharterBtn;

private javax.swing.JFrame caffeineFrame;

private javax.swing.JButton chartBtn;

private javax.swing.JFrame chartFrame;

private javax.swing.JButton chocBtn;

private javax.swing.JButton coffeeBtn;

private javax.swing.JButton cokeBtn;

private javax.swing.JButton glassWaterBtn;

private javax.swing.JLabel hrsLabel;

private javax.swing.JDialog hydrateDialog;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton4;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

private javax.swing.JLabel jLabel17;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JScrollPane jScrollPane4;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JLabel l1;

private javax.swing.JLabel l10;

private javax.swing.JLabel l11;

private javax.swing.JLabel l12;

private javax.swing.JLabel l13;

private javax.swing.JLabel l14;

private javax.swing.JLabel l2;

private javax.swing.JLabel l3;

private javax.swing.JLabel l4;

private javax.swing.JLabel l5;

private javax.swing.JLabel l6;

private javax.swing.JLabel l7;

private javax.swing.JLabel l8;

private javax.swing.JLabel l9;

private javax.swing.JLabel minLabel;

private javax.swing.JProgressBar p1;

private javax.swing.JProgressBar p10;

private javax.swing.JProgressBar p11;

private javax.swing.JProgressBar p12;

private javax.swing.JProgressBar p13;

private javax.swing.JProgressBar p14;

private javax.swing.JProgressBar p2;

private javax.swing.JProgressBar p3;

private javax.swing.JProgressBar p4;

private javax.swing.JProgressBar p5;

private javax.swing.JProgressBar p6;

private javax.swing.JProgressBar p7;

private javax.swing.JProgressBar p8;

private javax.swing.JProgressBar p9;

private javax.swing.JTextArea screenTimeAdviceTA;

private javax.swing.JButton screenTimeButton;

private javax.swing.JFrame screenTimeFrame;

private javax.swing.JLabel secLabel;

private javax.swing.JButton teaBtn;

private javax.swing.JButton walkThanks;

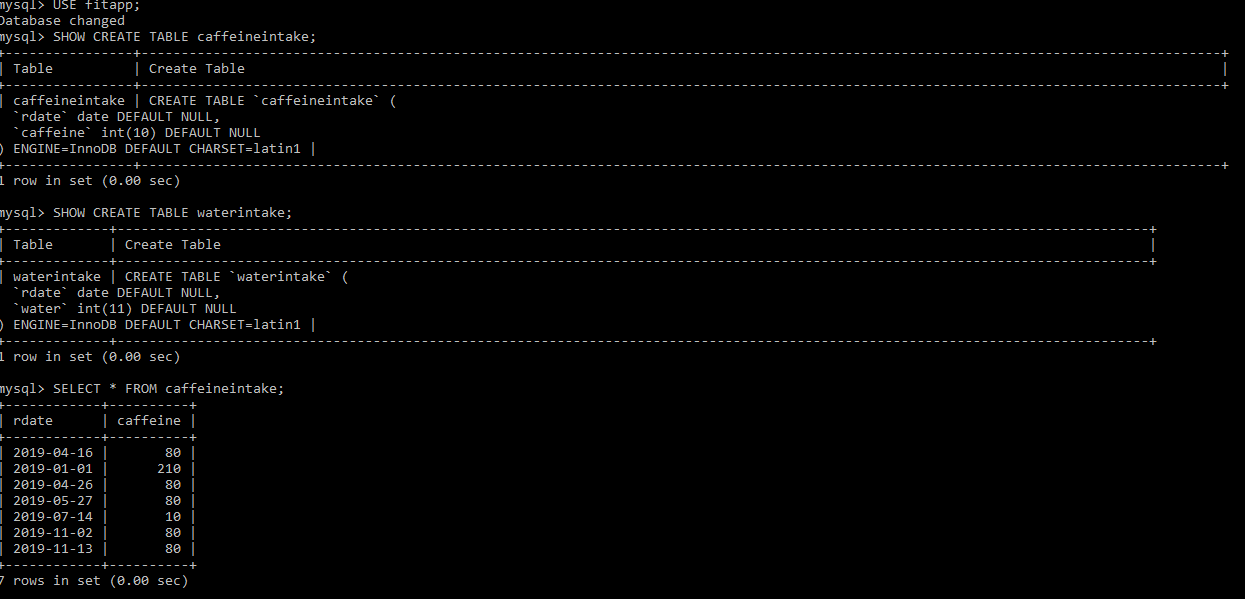
private javax.swing.JButton waterCharterBtn;

private javax.swing.JFrame waterFrame;

private javax.swing.JButton waterIntakeBtn;

// End of variables declaration

}

**MySQL DATABASE**

