**A PROJECT**

**ON**

**GYM MANAGEMENT SYSTEM**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENT**

**FOR THE CBSE INFORMATICS PRACTICES COURSE**

**FOR ACADEMIC YEAR 2017-2018**

**SUBMITTED BY**

(PRAJAY MEHTA)

(Roll No.: 08)

(XII SCIENCE-A)

**TO**

**CHANDRESH KATHIRIYA**

**INFORMATICS PRACTICES FACULTY**

**RADIANT ENGLISH ACADEMY**

**SURAT, GUJARAT**

**CERTIFICATE**

This is to certify that Mr PRAJAY MEHTA of Class XII SCIENCE – A of RADIANT ENGLISH ACADEMY, Surat has completed his project file under my supervision. He has taken proper care and shown utmost sincerity in the completion of this project.

I certify that this project is up to my expectation and as per the guidelines issued by CBSE.

Mr. Chandresh Kathiriya External Examiner

Signature

Principal Signature

**DECLARATION**

I declare that this project entitled “GYM MANAGEMENT SYATEM” done at RADIANT ENGLISH ACADEMY is a record of project work submitted by me for the partial fulfilment of the 10+2 of CBSE under the supervision and guidance of Mr. Chandresh Kathiriya.

The Project is genuine and not a reproduction of any project previously done or submitted.

(PRAJAY MEHTA)

(Roll No.: 08)

(XII SCIENCE A)

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher Mr. Chandresh Kathiriya as well as our Principal Mr. Trushar Parmar who gave me the golden opportunity to do this wonderful project on the topic GYM MANAGEMENT SYSTEM, which also helped me in doing a lot of research and I came to know about so many new things. So, I am really thankful to them.

Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

(PRAJAY MEHTA)

(Roll No.: 08)

(XII SCIENCE A)

* **Project Submitted By:**

1. PRAJAY MEHTA
2. HEET PATEL
3. PRANJAL KAKADIA

INTRODUCTION TO THE PROJECT

This project is designed to facilitate a gymming and fitness center to automate its operations of keeping records and store them in form of a large and user friendly database further facilitating easy access to the personnel.

# OBJECTIVES OF THE PROJECT

# 

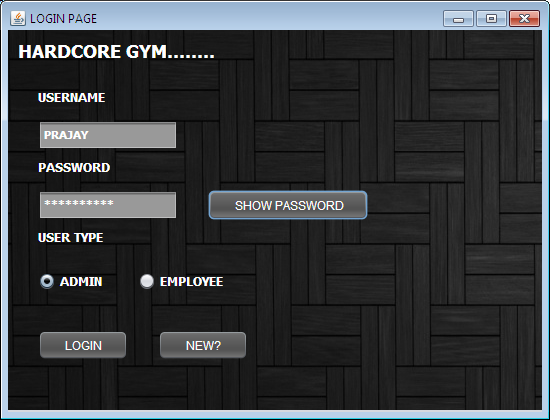
# What was the Problem?

* Existing system was manual.
* Time consuming as data entry which include calculations took lot of time.
* Searching was very complex as there could be 100’s of entry every year.
* The proposed system is expected to be faster than the existing system.
* The Project was made in order to effectively and efficiently caters to requirements of the fitness center. Very frequently the person who generally holds the tasks to manage the center needs to keep records of all the transactions as well as data manually. Generally, In order to structure these tasks Separate Registers are maintained. This whole process thus becomes quite cumbersome for them to control manually. Moreover, any wrong data entered mistakenly can bring serious results.
* This manually managed system of the store was also heavily prone to data loss due to certain causes misplacement of registers, destruction of registers, unauthorized access to registers etc. which can bring in disastrous consequences.
* The cost of maintenance of data and records of occurrence of transactions is very high.
* Searching a particular data specific to particular requirements is also very tedious in such system. In order to retrieve records. The responsible person needs to manually locate the appropriate register and locate the appropriate placement of that particular record which may be very time consuming.
* Data Redundancy is also a great issue in such kind of system. ”Redundancy” means repetition. Thus data modified or updated at a particular place may not be data modified or updated at the other related place which may create inconsistencies in data handling, destroys Data Integrity and creates confusion for the owner.
* **What the Software Provides in this Regard?**
* The software is capable enough to allow the concerned person to store and retrieve any type of record with just a single click of mouse. The software allows Interactive, Self describing Graphic User Interface environment where even standalone users can work very comfortably and easily.
* All the data pertaining to transactions or other important entities is kept at central database from where its attributes can be easily controlled. But, such kind of technical details are hidden from the standalone User. He just needs to type in correct details of the given entity and then click the save button with the help of mouse. However, that central repository of data can be easily accessed if required.
* Data Redundancy is no more the problem now. The data modified from one particular data entry form will reflect the modifications at the other related forms too. This has thus reduced the chances of data inconsistency in our data storage.
* There is no need to manage bulky registers now as data stored in the backend database can be readily retrieved either from the frontend form itself or directly from the database.
* Requires one time investment of setting up required Hardware and Software after which no more headaches are required by the Managers. Moreover, it also reduces dependence on Man Power.
* Effective Search measures are present at each and every data transactional forms from where by just entering a unique keyword for that data its whole records can be readily seen within microseconds. Moreover, Facility of Updating and Deleting of data through search is also available.

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **CONTENT** | **PAGE No.** |
| 1. | CERTIFICATE | 2 |
| 2. | DECLARATION | 3 |
| 3. | ACKNOWLEDGEMENT | 4 |
| 4. | INTRODUCTION | 6 |
| 5. | OBJECTIVES | 7 |
| 6. | ABOUT SOFTWARE | 9 |
| 7. | **FORMS:** |  |
| 8. | LOGIN FORM | 13 |
| 9. | NEW USER FORM | 17 |
| 10. | MAINPAGE FOR ADMIN | 20 |
| 11. | NEW MEMBER FORM | 25 |
| 12. | SEARCH AND UPDATE FORM | 33 |
| 13. | LIST OF MEMBERS FORM | 40 |
| 14. | VIEW PACKAGE FORM | 46 |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **CONTENT** | **PAGE No.** |
| 15. | VIEW BATCH FORM | 48 |
| 16. | VIEW TRAINERS FORM | 50 |
| 17. | ADD PACKAGE FORM | 53 |
| 18. | ADD BATCH FORM | 55 |
| 19. | ADD TRAINER FORM | 58 |
| 20. | CHANGE PASSWORD FORM | 60 |
| 21. | MAINPAGE FOR EMPLOYEE | 65 |
| 22. | TABLES | 66 |
| 23. | SCREEN SHOTS OF TABLES | 70 |
| 24. | FURTHER UPGRADATIONS | 72 |

**LOGIN PAGE:**

****

**CODE FOR INIT COMPONENTS:**

public login\_page() {

initComponents();

jLabel4.setVisible(false); // (jLabel4 is used for show

} password button.)

**CODE FOR LOGIN BUTTON:**

try

{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String useridfromuser=jTFusername.getText();

String passwordfromuser=jpassword.getText();

String user\_type=null;

if(jRBadmin.isSelected())

{

user\_type="ADMIN";

}

else if(jRBemployee.isSelected())

{

user\_type="EMPLOYEE";

}

String sql="select \* from login where username='"+useridfromuser+"' AND PASSWORD='"+passwordfromuser+"' AND USER\_TYPE='"+user\_type+"';";

ResultSet rs=stmt.executeQuery(sql);

if(rs.next())

{

if(jRBadmin.isSelected())

{

new mainpage().setVisible(true);

dispose();

}

else if(jRBemployee.isSelected())

{

new mainpage1().setVisible(true);

dispose();

}

}

else

{

JOptionPane.showMessageDialog(this,"check username or password");

}

}

catch (Exception e)

{

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

**CODE FOR NEW? BUTTON:**

private void signupActionPerformed(java.awt.event.ActionEvent evt)

{

new NEW\_USER().setVisible(true);

dispose();

}

**CODE FOR SHOW PASSWORD BUTTON:**

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

if(jLabel4.getText().equals(""))

{

jpassword.setEchoChar((char)0);

jLabel4.setText("Set");

}

else

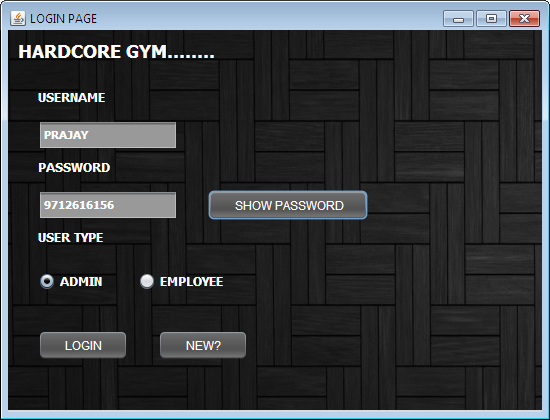
{

jpassword.setEchoChar('\*');

jLabel4.setText("");

}

}



**NEW USER FORM:**



**CODE FOR SIGN UP:**

try {

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String new\_username=jTFNewuser.getText();

String new\_password=jTFNewpass.getText();

String existing\_user=jTFExisting.getText();

String existing\_pass=jTFEpass.getText();

String user\_type=jComboBox2.getSelectedItem().toString();

String sql="select \* from login where username='"+existing\_user+"';";

ResultSet rs=stmt.executeQuery(sql);

//String password=rs.getString("password");

if(rs.next())

{

String Password=rs.getString("Password");

if(existing\_pass.equals(Password))

{

String sqlinsert="insert into login values('"+new\_username+"','"+new\_password+"','"+user\_type+"');";

stmt.executeUpdate(sqlinsert);

JOptionPane.showMessageDialog(this,"new user created....!!!");

jTFNewuser.setText(null);

jTFNewpass.setText(null);

jTFExisting.setText(null);

jTFEpass.setText(null);

}

else

{

JOptionPane.showMessageDialog(this,"check existing username or password");

}

}

}

catch (Exception e)

{

JOptionPane.showMessageDialog(this,e.getMessage());

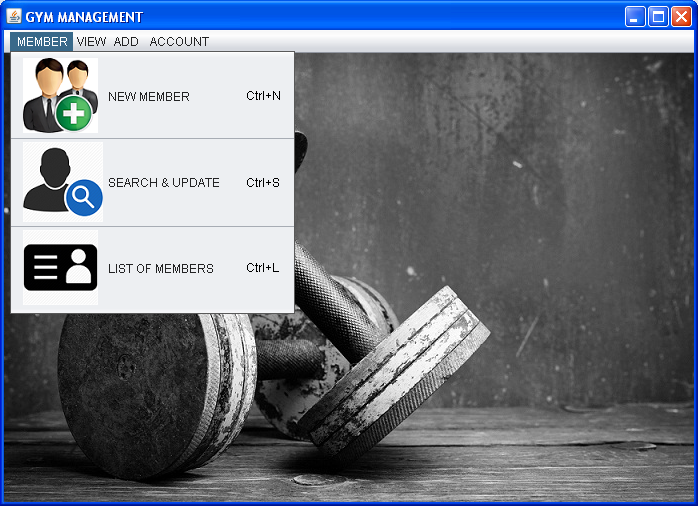
}

**CODE FOR BACK:**

new login\_page().setVisible(true);

dispose();

**MAINPAGE FOR ADMIN:**



**CODE FOR NEW MEMBER MENU:**

JFrame nf = new NEW\_MEMBER("User");

nf.setVisible(true);

dispose();

**CODE FOR SEARCH AND UPDATE MENU:**

JFrame nf = new SEARCH\_UPDATE("User");

nf.setVisible(true);

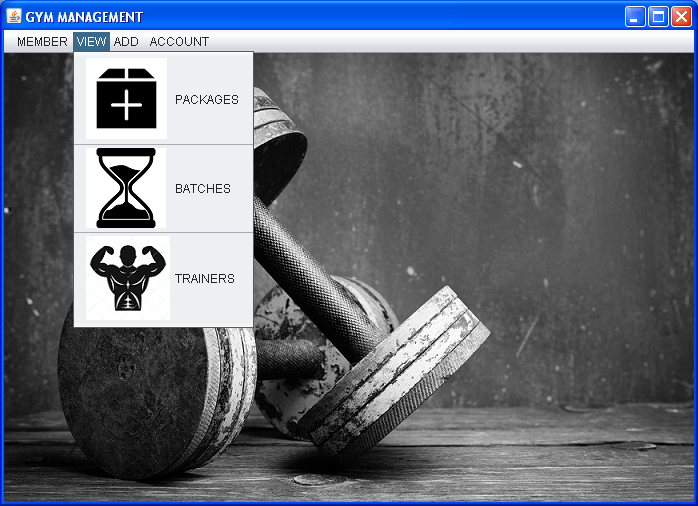
dispose();

**CODE FOR LIST OF MEMBERS MENU:**

JFrame nf = new LIST\_OF\_MEMBERS("User");

nf.setVisible(true);

dispose();



**CODE FOR PACKAGES MENU:**

JFrame nf = new VIEW\_PACKAGE("Admin");

nf.setVisible(true);

dispose();

**CODE FOR BATCHES MENU:**

JFrame nf = new VIEW\_BATCHES("Admin");

nf.setVisible(true);

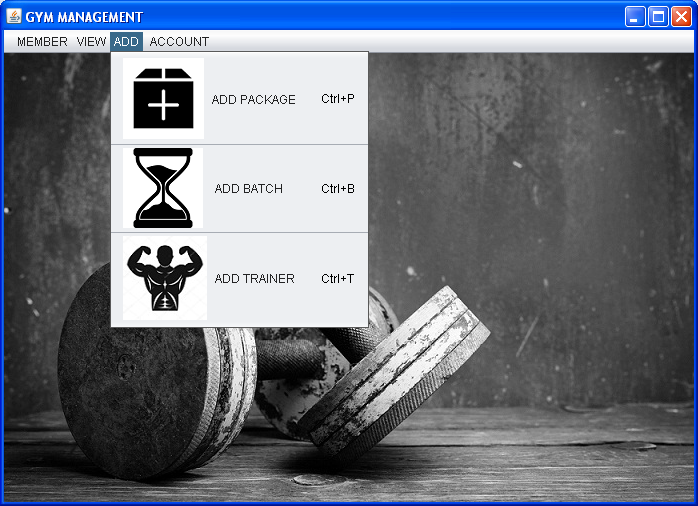
dispose();

**CODE FOR TRAINERS MENU:**

JFrame nf = new VIEW\_TRAINER("Admin");

nf.setVisible(true);

dispose();



**CODE FOR ADD PACKAGE MENU:**

new PACKAGE().setVisible(true);

dispose();

**CODE FOR ADD BATCH MENU:**

JFrame nf = new BATCHES("Admin");

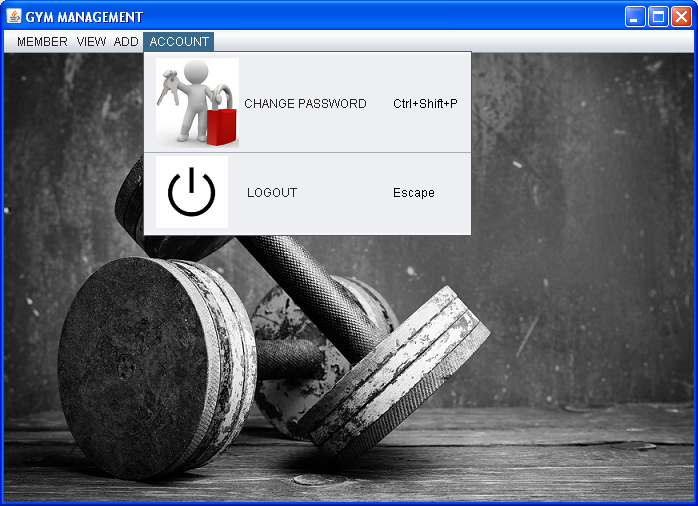
nf.setVisible(true);

dispose();

**CODE FOR ADD TRAINER MENU:**

new TRAINERS().setVisible(true);

dispose();



**CODE FOR CHANGE PASSWORD MENU:**

JFrame nf = new CHANGE\_PASSWORD("Admin");

nf.setVisible(true);

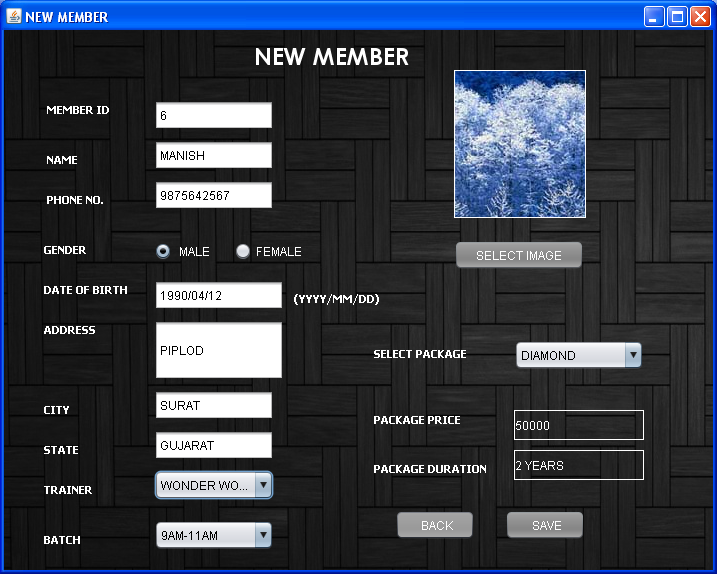
dispose();

**CODE FOR LOGOUT MENU:**

dispose();

new login\_page().setVisible(true);

**NEW MEMBER FORM**



**CODE FOR INIT COMPONENTS:**

String type = null; //(GLOBAL VARIABLE)

{

initComponents();

jFileChooser1.setVisible(false);

}

public NEW\_MEMBER(String usertype) {

initComponents();

jFileChooser1.setVisible(false);

type=usertype;

try

{

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

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="SELECT DISTINCT name FROM trainer;";

ResultSet rs=stmt.executeQuery(sql);

while(rs.next())

{

jComboBox3.addItem(rs.getString(1));

}

String sql1="SELECT DISTINCT batch FROM batch;";

ResultSet rs1=stmt.executeQuery(sql1);

while(rs1.next())

{

jComboBox4.addItem(rs1.getString(1));

}

String sql2="SELECT DISTINCT package FROM package;";

ResultSet rs2=stmt.executeQuery(sql2);

while(rs2.next())

{

jComboBox5.addItem(rs2.getString(1));

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

**CODE SELECT IMAGE BUTTON:**

jFileChooser1.setVisible(true);

jFileChooser1.showOpenDialog(this);

File file = jFileChooser1.getSelectedFile();

jLabel16.setIcon(new ImageIcon(file.toString()));

jFileChooser1.setVisible(false);

**CODE FOR BACK BUTTON:**

* THIS BACK BUTTON HELPS US TO RETRACE OUR PATH TO THE MAINPAGE WE HAVE COME FROM. FOR EXAMPLE, IF WE OPENED VIEW TRAINERS FORM FROM ADMIN`S OR EMPLOYEE`S MAINPAGE THEN ON PRESSING BACK WE AGAIN REACH THE SAME MAINPAGE i.e., OF ADMIN OR EMPLOYEE RESPECTIVELY.

if(type.equals("Admin"))

{

new mainpage().setVisible(true);

}

else if(type.equals("User"))

{

new mainpage1().setVisible(true);

}

dispose();

**CODE FOR SAVE BUTTON:**

try{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="SELECT IFNULL(MAX(MEMBER\_ID),'0') FROM new\_member;";

ResultSet rs=stmt.executeQuery(sql);

if(rs.next())

{

int member\_id=Integer.parseInt(rs.getString(1));

member\_id++;

jTextField4.setText(""+member\_id);

String cname=jTextField5.getText();

String phno=jTextField6.getText();

String gender="";

if(jRadioButton1.isSelected())

{

gender="M";

}

else if(jRadioButton2.isSelected())

{

gender="F";

}

String date=jTextField3.getText();

String address=jTextField8.getText();

String state=jTextField1.getText();

String city=jTextField2.getText();

String trainer=jComboBox3.getSelectedItem().toString();

String batch=jComboBox4.getSelectedItem().toString();

String cpackage=jComboBox5.getSelectedItem().toString();

String sql1="INSERT INTO new\_member VALUES('"+member\_id+"','"+cname+"','"+phno+"','"+gender+"','"+date+"','"+address+"','"+state+"','"+city+"','"+trainer+"','"+batch+"','"+cpackage+"');";

stmt.executeUpdate(sql1);

JOptionPane.showMessageDialog(this, "NEW MEMBER DETAILS ENTERED SUCCESSFULLY");

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

jTextField4.setText("");

jTextField5.setText("");

jTextField6.setText("");

jTextField7.setText("");

jTextField8.setText("");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

BufferedImage image;

File file = jFileChooser1.getSelectedFile();

try {

image = ImageIO.read(file);

ImageIO.write(image, "jpg",new File("G:\\PRAJAY\\PROJECT 8888\\PICTURES NEW MEMBER"+ file.getName()));

JOptionPane.showMessageDialog(this, "Image Saved Successfully...!");

jLabel6.setText("");

}

catch(IOException ex)

{

JOptionPane.showMessageDialog(this, ex.getMessage());

}

**CODE FOR SELECT PACKAGE COMBOBOX:**

try{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String trainer=jComboBox5.getSelectedItem().toString();

String sql="SELECT PRICE FROM PACKAGE WHERE PACKAGE='"+trainer+"';";

ResultSet rs=stmt.executeQuery(sql);

if(rs.next())

{

jLabel14.setText(rs.getString(1)+"");

}

String sql1="SELECT duration FROM PACKAGE WHERE PACKAGE='"+trainer+"';";

ResultSet rs1=stmt.executeQuery(sql1);

if(rs1.next())

{

jLabel15.setText(rs1.getString(1)+"");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**CODE FOR AUTONUMBERING OF MEMBER ID:**

try{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="SELECT IFNULL(MAX(MEMBER\_ID),'0') FROM new\_member;";

ResultSet rs=stmt.executeQuery(sql);

if(rs.next())

{

int member\_id=Integer.parseInt(rs.getString(1));

member\_id++;

jTextField4.setText(""+member\_id);

}

}

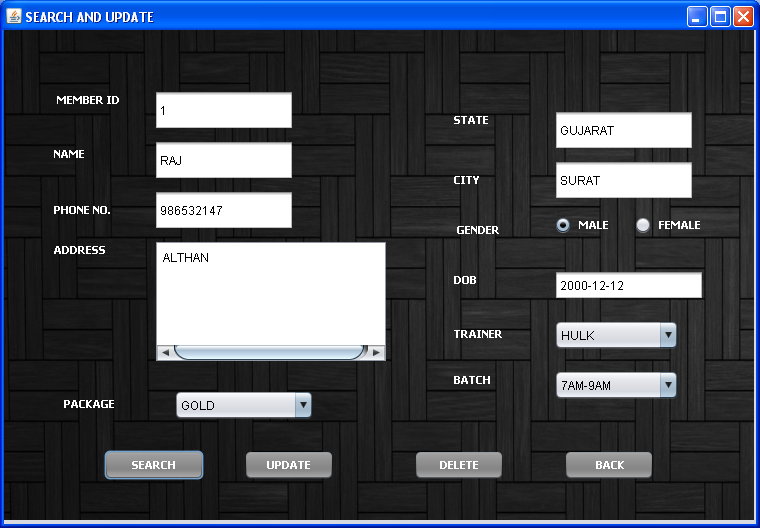
catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**SEARCH AND UPDATE FORM**



**CODE FOR SEARCH BUTTON:**

try

{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

int id=Integer.parseInt(jTextField1.getText());

String sql="select \* from new\_member where member\_id="+id+";";

ResultSet rs=stmt.executeQuery(sql);

if(rs.next())

{

jTextField2.setText(rs.getString("name"));

jTextField3.setText(rs.getString("phone\_no"));

jTextArea1.setText(rs.getString("address"));

jTextField4.setText(rs.getString("state"));

jTextField5.setText(rs.getString("city"));

jTextField6.setText(rs.getString("dob"));

String gender = rs.getString("Gender");

if(gender.equals("M"))

jRadioButton1.setSelected(true);

else if(gender.equals("F"))

jRadioButton2.setSelected(true);

jComboBox1.setSelectedItem(rs.getString("trainer"));

jComboBox3.setSelectedItem(rs.getString("package"));

jComboBox2.setSelectedItem(rs.getString("batch"));

}

else

{

JOptionPane.showMessageDialog(this, "MemberID not assigned...!!");

jTextField1.setText("");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**CODE FOR UPDATE BUTTON:**

try{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

int member\_id=Integer.parseInt(jTextField1.getText());

String name=jTextField2.getText();

String phone\_no=jTextField3.getText();

String address=jTextArea1.getText();

String state=jTextField4.getText();

String city=jTextField5.getText();

String gender=null;

if(jRadioButton1.isSelected())

{

gender="M";

}

else if(jRadioButton2.isSelected())

{

gender="F";

}

String DOB=jTextField6.getText();

String trainer=jComboBox1.getSelectedItem().toString();

String batch=jComboBox2.getSelectedItem().toString();

String cpackage=jComboBox3.getSelectedItem().toString();

String sql = "UPDATE NEW\_MEMBER SET name='"+name+"',phone\_no='"+phone\_no+"',gender='"+gender+"',dob='"+DOB+"',address='"+address+"',state='"+state+"',city='"+city+"',trainer='"+trainer+"',batch='"+batch+"',package='"+cpackage+"' where member\_id='"+member\_id+"';";

stmt.executeUpdate(sql);

JOptionPane.showMessageDialog(this, "Record successfully updated");

jTextArea1.setText("");

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

jTextField4.setText("");

jTextField5.setText("");

jTextField6.setText("");

buttonGroup1.clearSelection();

}

catch(Exception a){

JOptionPane.showMessageDialog(this, "Database Connection Error.....!");

}

**CODE FOR DELETE BUTTON:**

try{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String member\_id=jTextField1.getText();

String sql="DELETE from new\_member where member\_id="+member\_id+";";

stmt.executeUpdate(sql);

JOptionPane.showMessageDialog(this, "Record successfully deleted");

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

jTextField4.setText("");

jTextField5.setText("");

jTextField6.setText("");

buttonGroup1.clearSelection();

jTextArea1.setText("");

}

catch(Exception a){

JOptionPane.showMessageDialog(this, "Database Connection Error.....!");

}

**CODE FOR BACK BUTTON:**

if(type.equals("Admin"))

{

new mainpage().setVisible(true);

}

else if(type.equals("User"))

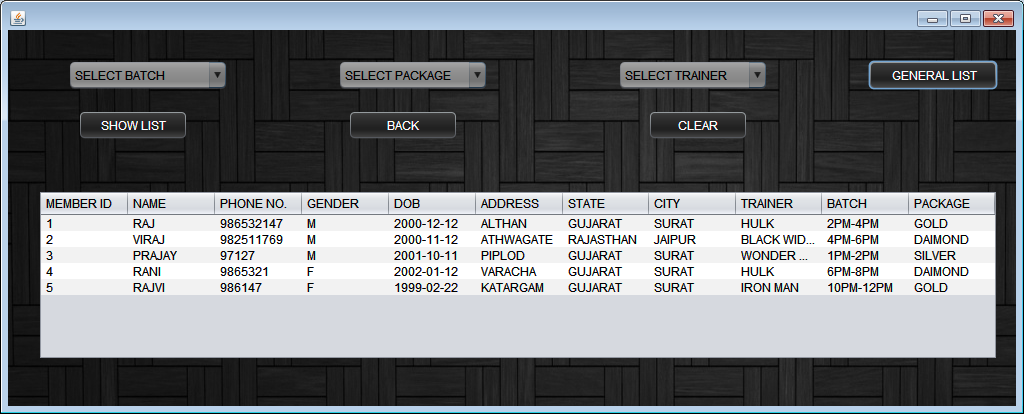
{

new mainpage1().setVisible(true);

}

dispose();

**LIST OF MEMBER FORM**



**CODE FOR INIT COMPONENTS:**

initComponents();

type=usertype;

try

{

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

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="SELECT DISTINCT batch FROM new\_member;";

ResultSet rs=stmt.executeQuery(sql);

while(rs.next())

{

jComboBoxBATCH.addItem(rs.getString(1));

}

String sql1="SELECT DISTINCT package FROM new\_member;";

ResultSet rs1=stmt.executeQuery(sql1);

while(rs1.next())

{

jComboBoxPACKAGE.addItem(rs1.getString(1));

}

String sql2="SELECT DISTINCT trainer FROM new\_member;";

ResultSet rs2=stmt.executeQuery(sql2);

while(rs2.next())

{

jComboBoxTRAINER.addItem(rs2.getString(1));

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**CODE FOR GENERAL LIST:**

* GENERAL LIST DOES NOT REQUIRE SELECTIO OF COMBOBOX. i.e. IT WOULD RUN EVEN IF WE DO NOT SELECT VALUES FROM COMBOBOX. ON PRESSING GENERAL LIST THE TABLE WOULD SHOW EACH AND EVERY RECORD PRESENT IN THE DATABASE.

DefaultTableModel model=(DefaultTableModel)jTable1.getModel();

try{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="select \* from new\_member;";

ResultSet rs=stmt.executeQuery(sql);

int rows=model.getRowCount();

if(rows>0)

{

for (int i=0;i<rows;i++)

{

model.removeRow(0);

}

}

while(rs.next())

{

model.addRow(new Object[] {rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),rs.getString(8),rs.getString(9),rs.getString(10),rs.getString(11)});

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**CODE FOR BACK BUTTON:**

if(type.equals("Admin"))

{

new mainpage().setVisible(true);

}

else if(type.equals("User"))

{

new mainpage1().setVisible(true);

}

dispose();

**CODE FOR SHOW LIST BUTTON:**

* FOR THE BUTTON SHOW LIST TO FUNCTION PROPERLY THE USER MUST SELECT ALL THE THREE COMBOBOXES OTHERWISE THE HE/SHE WOULD GET WILL BE AMBIGIOUS.

DefaultTableModel model=(DefaultTableModel)jTable1.getModel();

try{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String batch=jComboBoxBATCH.getSelectedItem().toString();

String cPackage=jComboBoxPACKAGE.getSelectedItem().toString();

String trainer=jComboBoxTRAINER.getSelectedItem().toString();

String sql="select \* from new\_member where batch='"+batch+"' and package='"+cPackage+"' and trainer='"+trainer+"';";

ResultSet rs=stmt.executeQuery(sql);

int rows=model.getRowCount();

if(rows>0)

{

for (int i=0;i<rows;i++)

{

model.removeRow(0);

}

}

while(rs.next())

{

model.addRow(new Object[] {rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),rs.getString(8),rs.getString(9),rs.getString(10),rs.getString(11)});

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**CODE FOR CLEAR BUTTON:**

DefaultTableModel model=(DefaultTableModel)jTable1.getModel();

int rows=model.getRowCount();

if(rows>0)

{

for (int i=0;i<rows;i++)

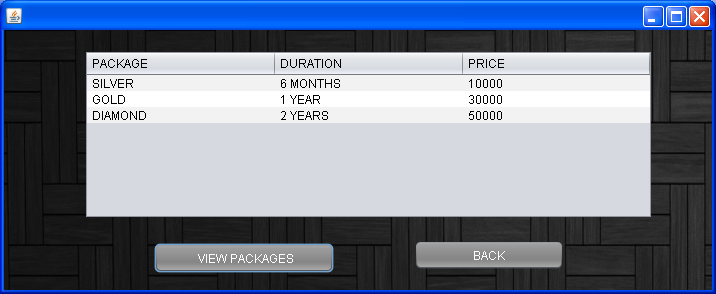
{

model.removeRow(0);

}

}

**VIEW PACKAGE FORM:**



DefaultTableModel model=(DefaultTableModel)jTable1.getModel();

try{

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

Connection conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="select Package,Duration,Price from package;";

ResultSet rs=stmt.executeQuery(sql);

int rows=model.getRowCount();

if(rows>0)

{

for (int i=0;i<rows;i++)

{

model.removeRow(0);

}

}

while(rs.next()){

model.addRow(new Object[]{rs.getString(1),rs.getString(2),rs.getString(3)});

}

}

catch(Exception e){

JOptionPane.showMessageDialog(this, "Database Connection Error.....!");

}

**CODE FOR BACK BUTTON:**

if(type.equals("Admin"))

{

new mainpage().setVisible(true);

}

else if(type.equals("User"))

{

new mainpage1().setVisible(true);

}

dispose();

**VIEW BATCH FORM:**



**CODE FOR VIEW BATCHES BUTTON:**

DefaultTableModel model=(DefaultTableModel)jTable1.getModel();

try{

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

Connection conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="select Batch,Strength from Batch;";

ResultSet rs=stmt.executeQuery(sql);

int rows=model.getRowCount();

if(rows>0)

{

for (int i=0;i<rows;i++)

{

model.removeRow(0);

}

}

while(rs.next()){

model.addRow(new Object[]{rs.getString(1),rs.getString(2)});

}

}

catch(Exception e){

JOptionPane.showMessageDialog(this, "Database Connection Error.....!");

}

**CODE FOR BACK BUTTON:**

if(type.equals("Admin"))

{

new mainpage().setVisible(true);

}

else if(type.equals("User"))

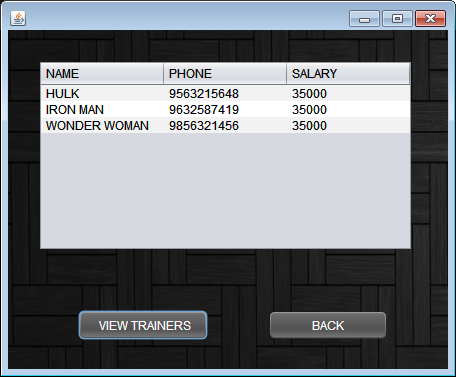
{

new mainpage1().setVisible(true);

}

dispose();

**VIEW TRAINERS FORM:**



**CODE FOR VIEW TRAINERS BUTTON:**

DefaultTableModel model=(DefaultTableModel)jTable1.getModel();

try{

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

Connection conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String sql="select Name,Phone,Salary from Trainer;";

ResultSet rs=stmt.executeQuery(sql);

int rows=model.getRowCount();

if(rows>0)

{

for (int i=0;i<rows;i++)

{

model.removeRow(0);

}

}

while(rs.next()){

model.addRow(new Object[]{rs.getString(1),rs.getString(2),rs.getString(3)}); }

}

catch(Exception e){

JOptionPane.showMessageDialog(this, "Database Connection Error.....!");

}

**CODE FOR BACK BUTTON:**

if(type.equals("Admin"))

{

new mainpage().setVisible(true);

}

else if(type.equals("User"))

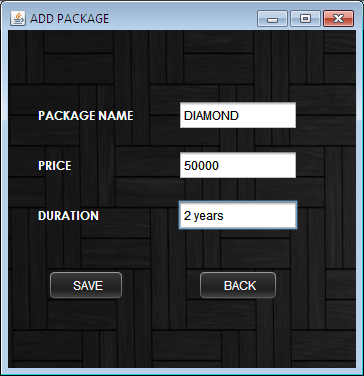
{

new mainpage1().setVisible(true);

}

dispose();

**ADD PACKAGE FORM**

****

**CODE FOR SAVE BUTTON:**

try

{

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

Connection conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

String Package\_name=jTFPackage.getText();

int ePrice=Integer.parseInt(jTFPrice.getText());

String Duration=jTFDuration.getText();

String sql = "INSERT INTO Package VALUES('"+Package\_name+"','"+ePrice+"','"+Duration+"');";

stmt.executeUpdate(sql);

JOptionPane.showMessageDialog(this, "Record successfully inserted");

jTFDuration.setText("");

jTFPackage.setText("");

jTFPrice.setText("");}

catch(Exception a)

{

JOptionPane.showMessageDialog(this, "Database Connection Error.....!");

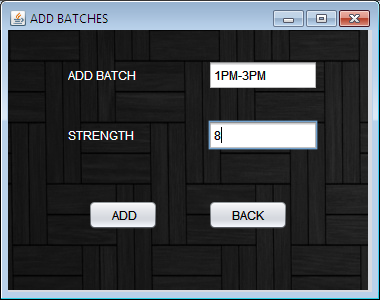
}

**CODE FOR BACK BUTTON:**

new mainpage().setVisible(true);

dispose();

**ADD BATCH FORM**

****

**CODE FOR ADD BUTTON:**

try{

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

Connection conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

int Strength=Integer.parseInt(jTFStrength.getText());

String Batch=jTFBatch.getText();

String sql = "INSERT INTO Batch VALUES('"+Batch+"','"+Strength+"')";

stmt.executeUpdate(sql);

JOptionPane.showMessageDialog(this, "Record successfully inserted");

jTFBatch.setText(null);

jTFStrength.setText(null);

}

catch(Exception a)

{

JOptionPane.showMessageDialog(this, "Database Connection Error.....!"); }

**CODE FOR BACK BUTTON:**

if(type.equals("Admin"))

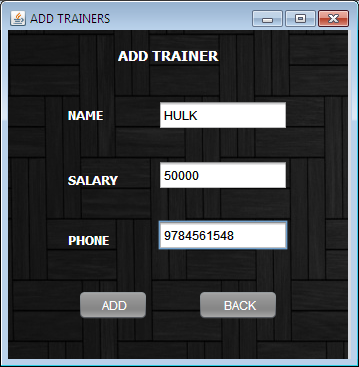
{ new mainpage().setVisible(true); }

else if(type.equals("User"))

{ new mainpage1().setVisible(true); }

dispose();

**ADD TRAINER FORM**



**CODE FOR ADD BUTTTON:**

try{

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

Connection conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement stmt=conn.createStatement();

int Salary=Integer.parseInt(jTFSalary.getText());

String Name=jTFName.getText();

int Phone=Integer.parseInt(jTFPhone.getText());

String sql = "INSERT INTO trainer VALUES('"+Name+"','"+Phone+"','"+Salary+"')";

stmt.executeUpdate(sql);

JOptionPane.showMessageDialog(this, "Record successfully inserted");

jTFName.setText(null);

jTFPhone.setText(null);

jTFSalary.setText(null);

}

catch(Exception a)

{

JOptionPane.showMessageDialog(this, "Database Connection Error.....!");

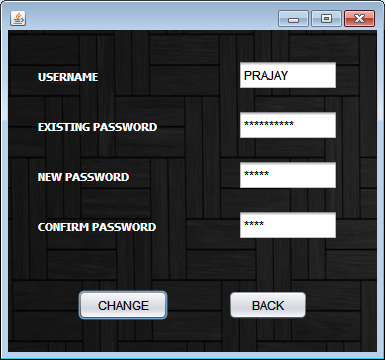
}

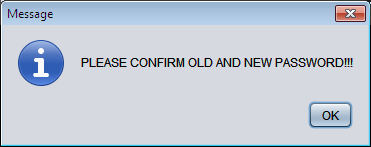
**CODE FOR BACK BUTTON:**

new mainpage().setVisible(true);

dispose();

**CHANGE PASSWORD FORM**

****



**CODE FOR CHANGE BUTTON:**

String uname=jTFUSERNAME.getText();

String strpass=jPF\_EXISTING\_PASSWORD.getText();

String newpass=jPF\_NEW\_PASSWORD.getText();

String conpass=jPF\_CONFIRM\_PASSWORD.getText();

try

{

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

Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root","admin");

Statement st=conn.createStatement();

ResultSet rs=st.executeQuery("Select username,password from login where username='"+uname+"';");

if(rs.next())

{

String usrname=rs.getString("username");

String passwd=rs.getString("password");

if(strpass.equals(passwd) && newpass.equals(conpass))

{

Statement st1=conn.createStatement();

String sql="UPDATE login SET password='"+newpass+"' where username='"+usrname+"';";

st1.executeUpdate(sql);

JOptionPane.showMessageDialog(null, "PASSWORD UPDATE SUCCESSFUL");

jTFUSERNAME.setText("");

jPF\_EXISTING\_PASSWORD.setText("");

jPF\_NEW\_PASSWORD.setText("");

jPF\_CONFIRM\_PASSWORD.setText("");

}

else if (uname.equals(null) || strpass.equals(null) || newpass.equals(null))

{

JOptionPane.showMessageDialog(null, "PLEASE ENTER ALL INFORMATION!!!");

}

else

{

JOptionPane.showMessageDialog(null, "PLEASE CONFIRM OLD AND NEW PASSWORD!!!");

}

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**CODE FOR BACK BUTTON:**

if(type.equals("Admin"))

{

new mainpage().setVisible(true);

}

else if(type.equals("User"))

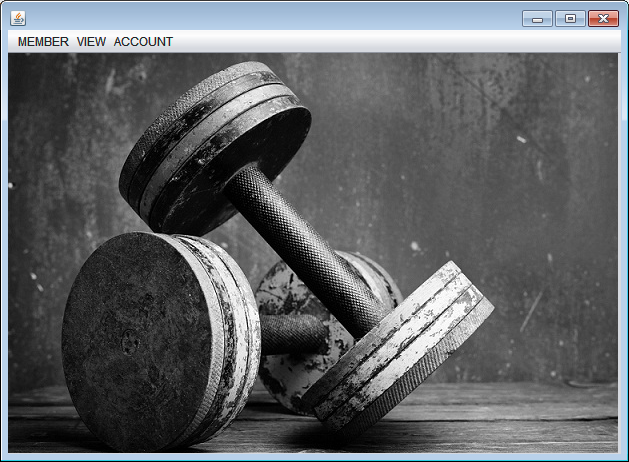
{

new mainpage1().setVisible(true);

}

dispose();

**MAINPAGE FOR EMPLOYEE:**



**MySQL TABLES:**

**CODES IN MYSQL:**

* Create table login(

username varchar(25),

password varchar(25),

user\_type varchar(25));

Insert into login values('heet','9586143361','ADMIN');

Insert into login values('prajay','9712616156','ADMIN');

Insert into login values('pranjal','0723','ADMIN');

Insert into login values('raju','007','EMPLOYEE');

* Create table trainer(

Name varchar(20),

Phone VARCHAR(20),

Salary integer(10));

INSERT INTO TRAINER VALUES('HULK','9563215648','35000');

INSERT INTO TRAINER VALUES('IRON MAN','9632587419','35000');

INSERT INTO TRAINER VALUES('WONDER WOMAN','9856321456','35000');

* Create table Batch(

Batch varchar(10),

Strength integer(10));

INSERT INTO BATCH VALUES('7AM-9AM','8');

INSERT INTO BATCH VALUES('9AM-11AM','8');

INSERT INTO BATCH VALUES('1PM-3PM','8');

* Create table Package(

Package varchar(20),

Price integer(11),

Duration varchar(10));

INSERT INTO PACKAGE VALUES('SILVER','10000','6 MONTHS');

INSERT INTO PACKAGE VALUES('GOLD','30000','1 YEAR');

INSERT INTO PACKAGE VALUES('DIAMOND','50000','2 YEARS');

* Create table new\_member(

member\_id integer(40) primary key,

name varchar(400),

phone\_no varchar(20),

gender char(1),

dob date,

address varchar(50),

state varchar(20),

city varchar(20),

trainer varchar(20),

batch varchar(20),

package varchar(20)

);

INSERT INTO NEW\_MEMBER VALUES('1','RAJ','986532147','M','2000/12/12','ALTHAN','GUJARAT','SURAT','HULK','2PM-4PM','GOLD');

INSERT INTO NEW\_MEMBER VALUES('2','VIRAJ','982511769','M','2000/11/12','ATHWAGATE','RAJASTHAN','JAIPUR','BLACK WIDOW','4PM-6PM','DAIMOND');

INSERT INTO NEW\_MEMBER VALUES('3','PRAJAY','97127','M','2001/10/11','PIPLOD','GUJARAT','SURAT','WONDER WOMEN','1PM-2PM','SILVER');

INSERT INTO NEW\_MEMBER VALUES('4','RANI','9865321','F','2002/1/12','VARACHA','GUJARAT','SURAT','HULK','6PM-8PM','DAIMOND');

INSERT INTO NEW\_MEMBER VALUES('5','RAJVI','986147','F','1999/2/22','KATARGAM','GUJARAT','SURAT','IRON MAN','10PM-12PM','GOLD');

**FURTHER UPGRADATIONS:**

* The system does not notify whether the batch is completely full and no other member can enroll themselves in that batch.

* The system does not count the number of days left / completed for the given package .it just shows the time period for a given package and price of that package.
* In list of member`s form the system cannot search on the basis of any one combo box. That is all the combo boxes must be selected in order to gain proper result.