

**OBJECT ORIENTED PROGRAMMING**

**CSL-210**

**Project Report**

**“COVID REGISTRATION MANAGEMENT”**

**Class: BS(CS)-2A**

**Course Teacher: Miss Sameena Javed**

**Lab Teacher: Miss Salas Akbar**

**Presented By:**

Umer Bahadur (02-134221-007)

**Table of Contents**

01. Introduction to system 3

02. Database Explanation 4

03. UML 5-6

04. Screenshot of forms 6-10

05. Code 11-56

06. Conclusion 57

1. **INTRODUCTION TO SYSTEM:**

COVID Vaccine Management System is one that organizes and saves peoples information electronically to meet the need of vaccinating staff. Both patients and staff benefit from the system, which allows them to keep track of their vaccine’s dose and next appointments. Vaccine management system is helpful in data sharing across recipients, care providers, hospitals, agencies, and local, state, and federal governments on one common platform.

Covid Vaccine Management System provides the following services:

**Welcome**: Welcome class welcomes user and allows user to choose between patient and admin login.

**Login form for admin**: Login form is used by admin only; you are asked to enter your id and password which is then checked from the database to match the credentials.

**Admin City:** Admin City is a JFrametable. Where user admin can add vaccines to the inventory of the City Hospital. He has options to add different vaccines different quantities and manufactures as required. He can edit the vaccines available as well by removing them.

**Admin Central:** Admin Central is a JFrametable. Where user admin can add vaccines to the inventory of the Central Hospital. He has options to add different vaccines different quantities and manufactures as required. He can edit the vaccines available as well by removing them.

**Patient form**: Patient form is used by the patients to enter their information. Patient is asked to enter his name, his father name, CNIC number and patient situation. The patient panned displays current date and time as well.

**Patient Panel:** Patient panel displays 2 buttons, City Hospital and Central Hospital from which user can choose anyone of them.

**City Hospital:** City Hospital is a JFrametable. Where user can book appointment. He has options to choose from different vaccines different timeslots available. This panel displays available vaccines from which user can choose from.

**Central Hospital:** Central Hospital is a JFrametable. Where user can book appointment. He has options to choose from different vaccines different timeslots available. This panel displays available vaccines from which user can choose from.

**Patient Slip**: Patient Slip displays a patient slip which displays the information patient selected while booking the appointment.

1. **DATABASE EXPLANATION:**

The class DBConnection consists of tables namely. Login, patient, vaccine, and vaccine1.

The method LoginExample connects the MS Access file to the java project with the help of Ucan acccess by which project loads data from the Access file .

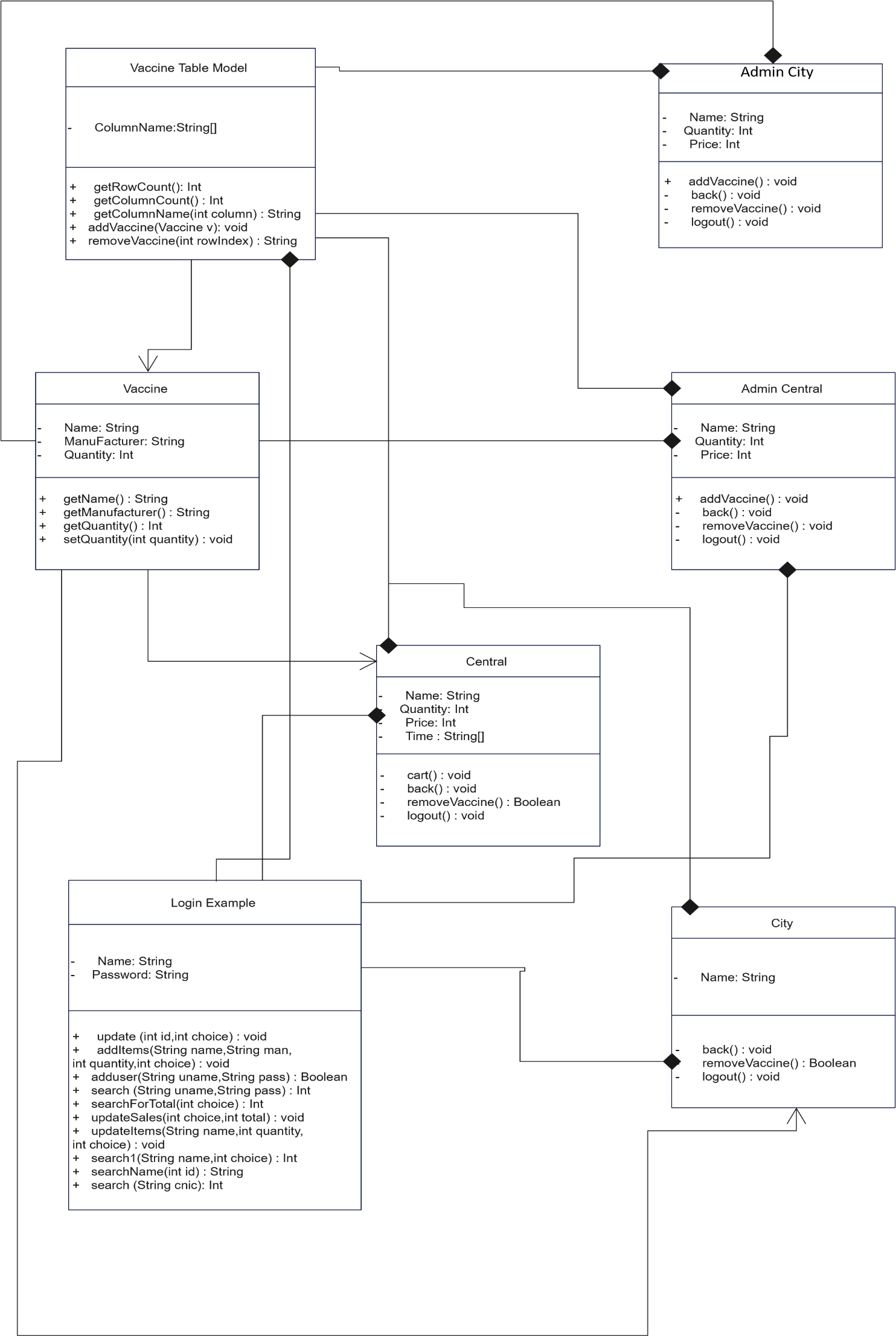
Login Table consist of (Admin) Username and Pasword. Which is called and read by the LoginPage() class, and the credentials are matched with the entered credentials.

Patient Data base Table : Patient consist of Patient data which is entered by the patient in the Module() class. All the data is stored in this table, whenever new data is entered it is first matched by the patient’s entered CNIC number if the CNIC number doesn’t exist Dost1 will be marked, if the CNIC already exists second dose will be marked and vaccine name will be registered.

The table ‘Vaccine’ contains the information of the vaccines for City Hospital that the admin orders and stores in the database.

The table ‘Vaccine1’ contains the information of the vaccines for Central Hospital that the admin orders and stores in the database.

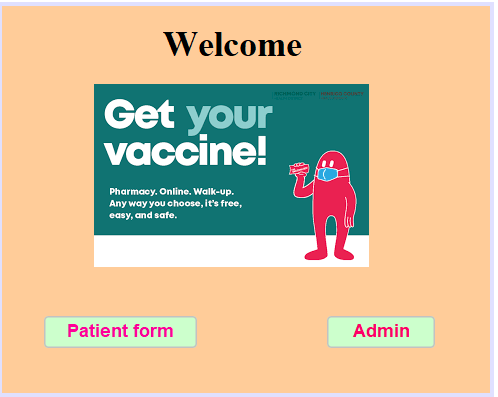
.

3. **UML DIAGRAM:**

1. **SCREENSHOT OF ALL FORMS:**

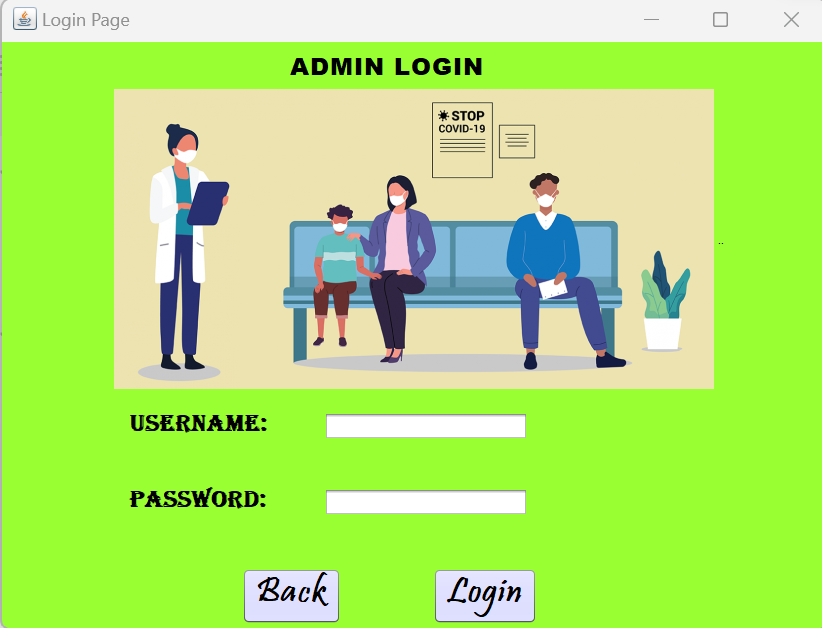
**WELCOME FORM:**

Welcome form allows you to click on either patient form or Admin login form.

**,**

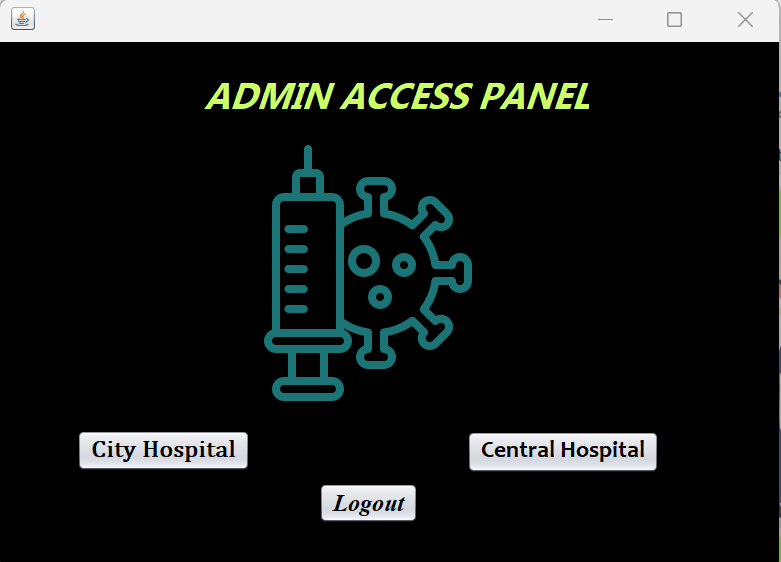
**ADMIN LOGIN FORM:**

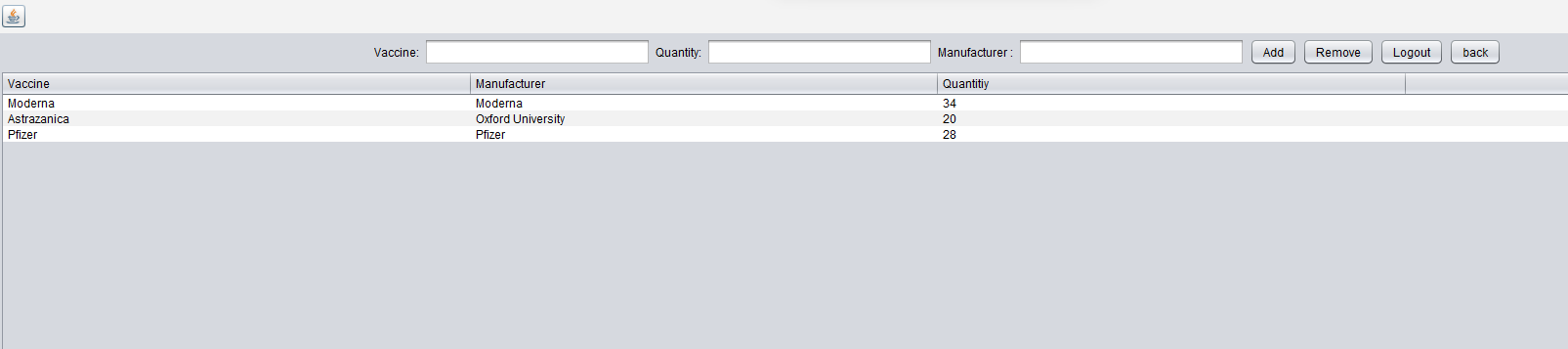
Admin login form: Login form need username and password to login.

****

**ADMIN ACCESS PANEL:**

After Admin login, Admin Access Panel will have two buttons which will allow admin to add vaccine in either one of the hospital.

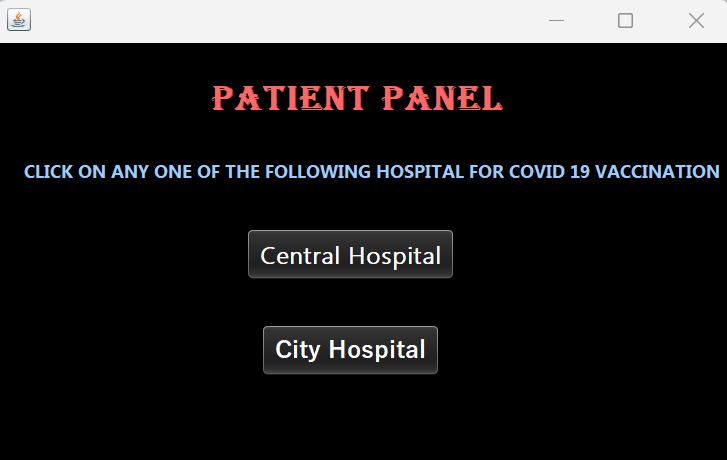
****

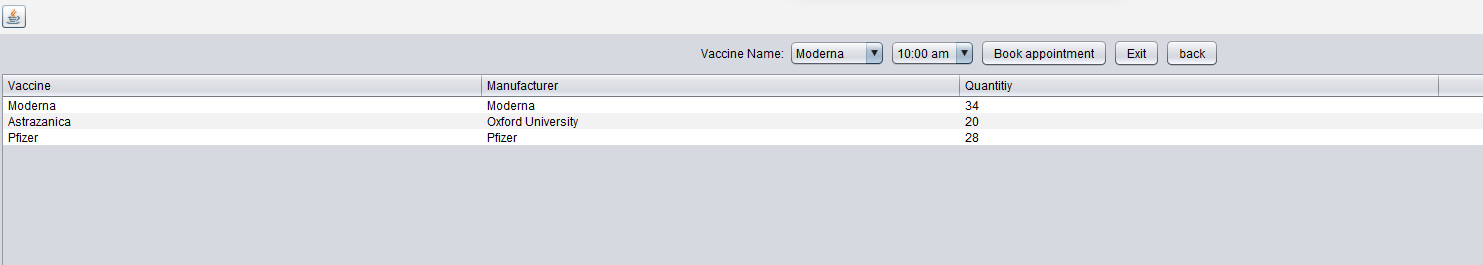
****

**PATIENT FORM:**

This form will take input from the user like Patient name, Father Name, CNIC number and Patient situation.

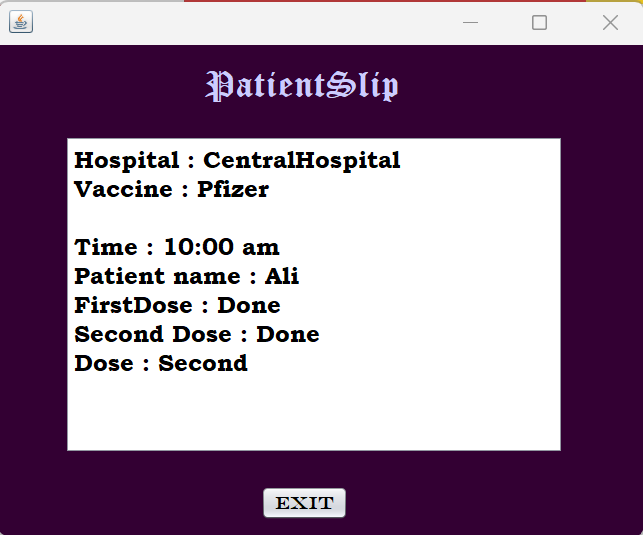
**PATIENT PANEL FORM :**

****

****

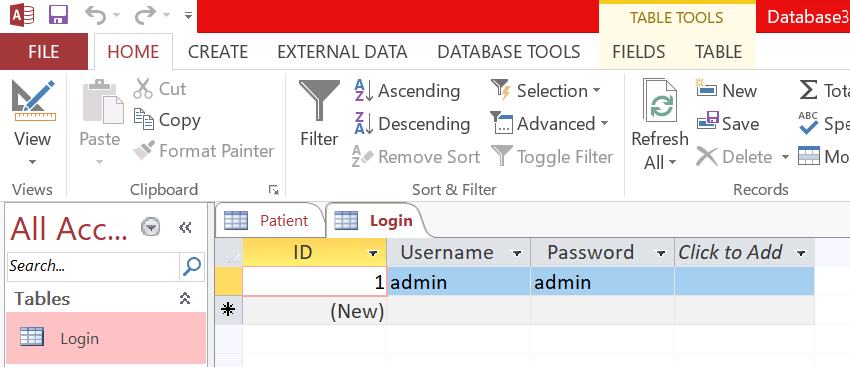
**PATIENT SLIP:**

It prints all the details like Name, Hospital, Vaccine, covid dose etc.

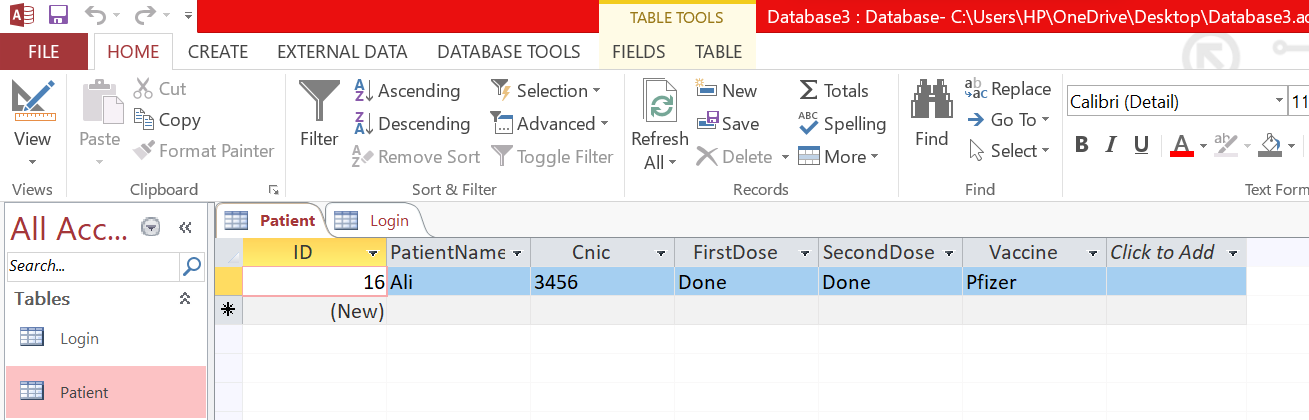


**DATABASE:**

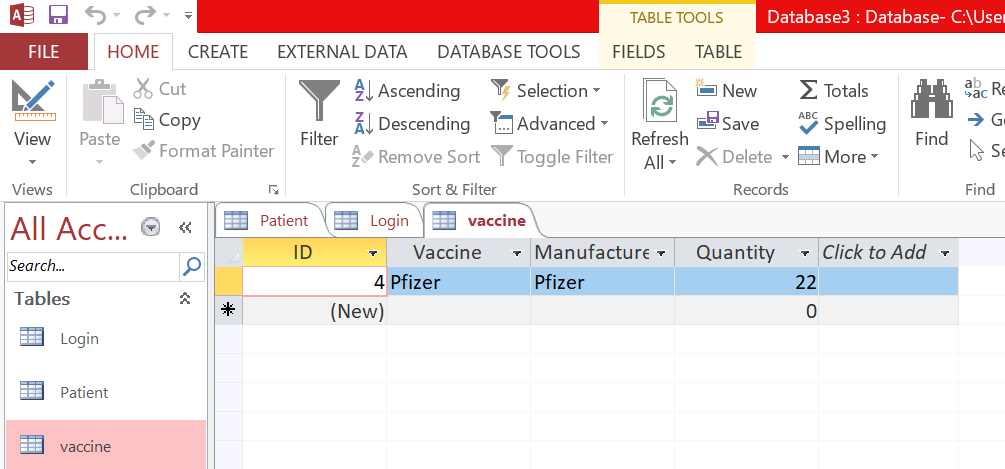
Admin Login Database:

****

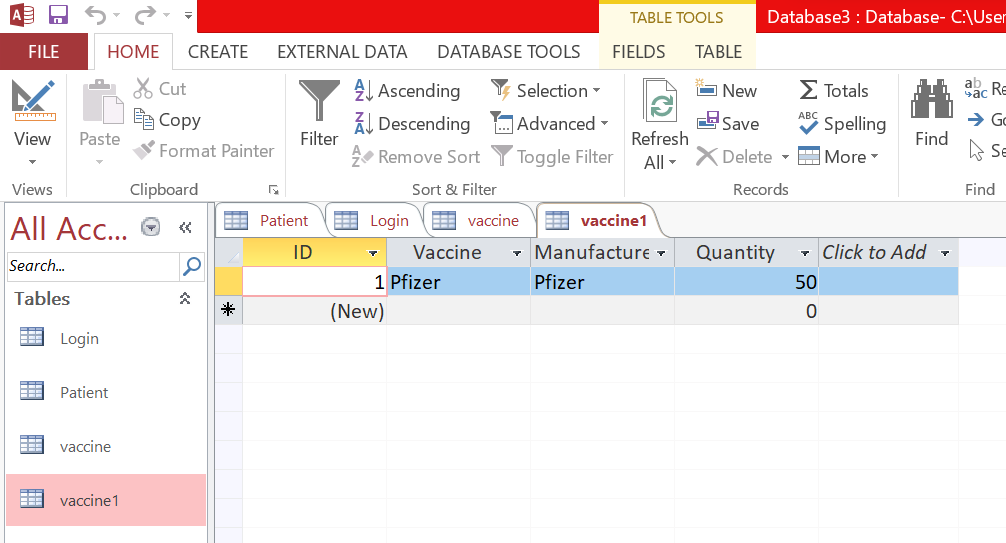
Patient Database:

****

Vaccine Database for City Hospital:

****

Vaccine Database for Central Hospital:

****

1. **CODE SNIPPETS:**

**AdminCity.java:**

package covidmanagementsystem;

import java.awt.\*;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

import java.util.ArrayList;

public class Admin extends JFrame {

// Declare GUI components

private JLabel lblVaccinename;

private JTextField txtVaccineName;

private JLabel lblQuantity;

private JLabel lblPrice;

private JTextField txtQuantity;

private JTextField txtPrice;

private JButton btnAdd;

private JButton btnRemove;

private JButton btnLogout;

private JTable tblInventory;

private JScrollPane scrollPane;

private JButton btnback;

// Declare data model for table

private VaccineTable tableModel;

public Admin() {

// ImageIcon icon = new ImageIcon("C:\\Users\\noorm\\Downloads\\1.png");

// Set up GUI components

lblVaccinename = new JLabel("Vaccine:");

txtVaccineName = new JTextField(20);

lblQuantity = new JLabel("Quantity:");

txtQuantity = new JTextField(20);

lblPrice=new JLabel("Manufacturer :");

txtPrice=new JTextField(20);

btnAdd = new JButton("Add");

btnRemove = new JButton("Remove");

btnLogout = new JButton("Logout");

btnback=new JButton("back");

// btnback.setIcon(icon);

LoginExample l1=new LoginExample();

try {

// Set up table and data model

tableModel =l1.getItems(1);

} catch (SQLException ex) {

Logger.getLogger(Admin.class.getName()).log(Level.SEVERE, null, ex);

}

tblInventory = new JTable(tableModel);

scrollPane = new JScrollPane(tblInventory);

// Set up layout

setLayout(new BorderLayout());

JPanel pnlInput = new JPanel(new FlowLayout());

pnlInput.add(lblVaccinename);

pnlInput.add(txtVaccineName);

pnlInput.add(lblQuantity);

pnlInput.add(txtQuantity);

pnlInput.add(lblPrice);

pnlInput.add(txtPrice);

pnlInput.add(btnAdd);

pnlInput.add(btnRemove);

pnlInput.add(btnLogout);

pnlInput.add(btnback);

add(pnlInput, BorderLayout.NORTH);

add(scrollPane, BorderLayout.CENTER);

// Set up event handlers

btnAdd.addActionListener(e -> {

try {

addVaccine();

} catch (SQLException ex) {

Logger.getLogger(Admin.class.getName()).log(Level.SEVERE, null, ex);

}

});

btnRemove.addActionListener(e -> removeVaccine());

btnLogout.addActionListener(e -> logout());

btnback.addActionListener(e->back());

//btnAdd.addActionListener(e->);

}

private void addVaccine() throws SQLException {

String name = txtVaccineName.getText();

int quantity = Integer.parseInt(txtQuantity.getText());

String man= (txtPrice.getText());

int selectedRow=0;

boolean v1=false;

for (int i = 0; i < tableModel.getRowCount(); i++) {

for (int j = 0; j < tableModel.getColumnCount(); j++) {

if (tableModel.getValueAt(i, 0).equals(name)) {

v1=true;

selectedRow = i;

break;

}

}

}

if (v1){

int q=Integer.parseInt(tableModel.getValueAt(selectedRow, 2).toString());

man=tableModel.getValueAt(selectedRow, 1).toString();

Vaccine vaccine = new Vaccine(name,man,quantity+q );

tableModel.addVaccine(vaccine);

txtPrice.setText("");

txtVaccineName.setText("");

txtQuantity.setText("");

tableModel.removeVaccine(selectedRow);

LoginExample l1=new LoginExample();

l1.updateItems(name, quantity+q, 1);

}

else{

// int q=Integer.parseInt(tableModel.getValueAt(selectedRow, 1).toString());

Vaccine vaccine = new Vaccine(name,man,quantity);

System.out.println(tableModel.getColumnCount());

tableModel.addVaccine(vaccine);

// User u1=new User();

//User.cmbFlowerName.addItem(name);

LoginExample l1=new LoginExample();

l1.addItems(name, man,quantity, 1);

txtPrice.setText("");

txtVaccineName.setText("");

txtQuantity.setText("");

}

}

private void removeVaccine() {

String s1=txtVaccineName.getText();

int selectedRow = tblInventory.getSelectedRow();

int q=Integer.parseInt(txtQuantity.getText());

//int price1=Integer.parseInt(txtPrice.getText());

//

for (int i = 0; i < tableModel.getRowCount(); i++) {

for (int j = 0; j < tableModel.getColumnCount(); j++) {

if (tableModel.getValueAt(i, 0).equals(s1)) {

selectedRow = i;

break;

}

}

}

if (selectedRow != -1) {

String man =(tableModel.getValueAt(selectedRow, 1).toString());

int quantity=Integer.parseInt(tableModel.getValueAt(selectedRow, 2).toString());

if (quantity<q){

JOptionPane.showMessageDialog(null, "Error");

}

else{

quantity-=q;

// tableModel.setValueAt(quantity, selectedRow, 2);

Vaccine f1=new Vaccine(s1, man,quantity);

tableModel.addVaccine(f1);

LoginExample l=new LoginExample();

try {

l.updateItems(s1,quantity,1);

} catch (SQLException ex) {

Logger.getLogger(Admin.class.getName()).log(Level.SEVERE, null, ex);

}

tableModel.removeVaccine(selectedRow);

txtPrice.setText("");

txtVaccineName.setText("");

txtQuantity.setText("");

//tableModel.setValueAt(quantity, selectedRow, 1);

}

}

}

private void back(){

AdminAcessPanel a1=new AdminAcessPanel();

setVisible(false);

a1.setVisible(true);

}

private void logout() {

LoginPage manager = new LoginPage();

manager.setVisible(true);

setVisible(false);

}

}

**ADMIN CENTRAL.JAVA:**

package covidmanagementsystem;

import java.awt.\*;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

import java.util.ArrayList;

public class AdminCentral extends JFrame {

// Declare GUI components

private JLabel lblFlowerName;

private JTextField txtVaccineName;

private JLabel lblQuantity;

private JLabel lblPrice;

private JTextField txtQuantity;

private JTextField txtPrice;

private JButton btnAdd;

private JButton btnRemove;

private JButton btnLogout;

private JTable tblInventory;

private JScrollPane scrollPane;

private JButton btnback;

// Declare data model for table

private VaccineTable tableModel;

public AdminCentral() {

// ImageIcon icon = new ImageIcon("C:\\Users\\noorm\\Downloads\\1.png");

// Set up GUI components

lblFlowerName = new JLabel("Vaccine:");

txtVaccineName = new JTextField(20);

lblQuantity = new JLabel("Quantity:");

txtQuantity = new JTextField(20);

lblPrice=new JLabel("Manufacturer :");

txtPrice=new JTextField(20);

btnAdd = new JButton("Add");

btnRemove = new JButton("Remove");

btnLogout = new JButton("Logout");

btnback=new JButton("back");

LoginExample l1=new LoginExample();

try {

// Set up table and data model

tableModel =l1.getItems(1);

} catch (SQLException ex) {

Logger.getLogger(Admin.class.getName()).log(Level.SEVERE, null, ex);

}

tblInventory = new JTable(tableModel);

scrollPane = new JScrollPane(tblInventory);

// Set up layout

setLayout(new BorderLayout());

JPanel pnlInput = new JPanel(new FlowLayout());

pnlInput.add(lblFlowerName);

pnlInput.add(txtVaccineName);

pnlInput.add(lblQuantity);

pnlInput.add(txtQuantity);

pnlInput.add(lblPrice);

pnlInput.add(txtPrice);

pnlInput.add(btnAdd);

pnlInput.add(btnRemove);

pnlInput.add(btnLogout);

pnlInput.add(btnback);

add(pnlInput, BorderLayout.NORTH);

add(scrollPane, BorderLayout.CENTER);

// Set up event handlers

btnAdd.addActionListener(e -> {

try {

addVaccine();

} catch (SQLException ex) {

Logger.getLogger(Admin.class.getName()).log(Level.SEVERE, null, ex);

}

});

btnRemove.addActionListener(e -> removeVaccine());

btnLogout.addActionListener(e -> logout());

btnback.addActionListener(e->back());

}

private void addVaccine() throws SQLException {

String name = txtVaccineName.getText();

int quantity = Integer.parseInt(txtQuantity.getText());

String man= (txtPrice.getText());

int selectedRow=0;

boolean v1=false;

for (int i = 0; i < tableModel.getRowCount(); i++) {

for (int j = 0; j < tableModel.getColumnCount(); j++) {

if (tableModel.getValueAt(i, 0).equals(name)) {

v1=true;

selectedRow = i;

break;

}

}

}

if (v1){

int q=Integer.parseInt(tableModel.getValueAt(selectedRow, 2).toString());

man=tableModel.getValueAt(selectedRow, 1).toString();

Vaccine vaccine = new Vaccine(name,man,quantity+q );

tableModel.addVaccine(vaccine);

txtPrice.setText("");

txtVaccineName.setText("");

txtQuantity.setText("");

tableModel.removeVaccine(selectedRow);

LoginExample l1=new LoginExample();

l1.updateItems(name, quantity+q, 2);

}

else{

// int q=Integer.parseInt(tableModel.getValueAt(selectedRow, 1).toString());

Vaccine flower = new Vaccine(name,man,quantity);

System.out.println(tableModel.getColumnCount());

tableModel.addVaccine(flower);

// User u1=new User();

//User.cmbFlowerName.addItem(name);

LoginExample l1=new LoginExample();

l1.addItems(name, man,quantity, 2);

txtPrice.setText("");

txtVaccineName.setText("");

txtQuantity.setText("");

}

}

private void removeVaccine() {

String s1=txtVaccineName.getText();

int selectedRow = tblInventory.getSelectedRow();

int q=Integer.parseInt(txtQuantity.getText());

//int price1=Integer.parseInt(txtPrice.getText());

//

for (int i = 0; i < tableModel.getRowCount(); i++) {

for (int j = 0; j < tableModel.getColumnCount(); j++) {

if (tableModel.getValueAt(i, 0).equals(s1)) {

selectedRow = i;

break;

}

}

}

if (selectedRow != -1) {

String man =(tableModel.getValueAt(selectedRow, 1).toString());

int quantity=Integer.parseInt(tableModel.getValueAt(selectedRow, 2).toString());

if (quantity<q){

JOptionPane.showMessageDialog(null, "Error");

}

else{

quantity-=q;

// tableModel.setValueAt(quantity, selectedRow, 2);

Vaccine f1=new Vaccine(s1, man,quantity);

tableModel.addVaccine(f1);

tableModel.removeVaccine(selectedRow);

LoginExample l=new LoginExample();

try {

l.updateItems(s1,quantity,2);

} catch (SQLException ex) {

Logger.getLogger(AdminCentral.class.getName()).log(Level.SEVERE, null, ex);

}

txtPrice.setText("");

txtVaccineName.setText("");

txtQuantity.setText("");

//tableModel.setValueAt(quantity, selectedRow, 1);

}

}

}

private void back(){

AdminAcessPanel a1=new AdminAcessPanel();

setVisible(false);

a1.setVisible(true);

}

private void logout() {

LoginPage manager = new LoginPage();

manager.setVisible(true);

setVisible(false);

}

}

**Central.java:**

package covidmanagementsystem;

import java.awt.\*;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

public class Central extends JFrame {

// Declare GUI components

private JLabel lblVaccineName;

public static JComboBox cmbVaccineName;

private JLabel lblQuantity;

private JComboBox txtQuantity;

private JButton btnOrder;

private JButton btnLogout;

private JTable tblOrders;

private JScrollPane scrollPane;

private JButton viewCart;

private JButton btnback;

public String var;

// Declare data model for table

//private OrderTableModel tableModel;

private VaccineTable tablemodel1;

// public String[]it;

public static ArrayList<String> arr;

public Central() {

var=new String();

// Set up GUI components

LoginExample l1=new LoginExample();

arr=new ArrayList<>();

String []it={};

try {

it=l1.getVaccines(2);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

lblVaccineName = new JLabel("Vaccine Name:");

cmbVaccineName = new JComboBox(it);

// lblQuantity = new JLabel("Quantity:");

String[]time={"10:00 am","12:00 pm","2:00 pm"};

txtQuantity = new JComboBox(time);

btnOrder = new JButton("Book appointment");

btnLogout = new JButton("Exit");

// viewCart=new JButton("View Cart");

btnback =new JButton("back");

// LoginExample l1=new LoginExample();

// tableModel = new OrderTableModel();

try {

// Set up table and data model

tablemodel1=l1.getItems(2);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

tblOrders = new JTable(tablemodel1);

scrollPane = new JScrollPane(tblOrders);

// Set up layout

setLayout(new BorderLayout());

JPanel pnlInput = new JPanel(new FlowLayout());

pnlInput.add(lblVaccineName);

pnlInput.add(cmbVaccineName);

//pnlInput.add(lblQuantity);

pnlInput.add(txtQuantity);

pnlInput.add(btnOrder);

pnlInput.add(btnLogout);

// pnlInput.add(viewCart);

pnlInput.add(btnback);

add(pnlInput, BorderLayout.NORTH);

add(scrollPane, BorderLayout.CENTER);

// Set up event handlers

btnback.addActionListener(e->back());

btnOrder.addActionListener(e -> {

try {

bookAppointment();

} catch (SQLException ex) {

Logger.getLogger(City.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(City.class.getName()).log(Level.SEVERE, null, ex);

}

});

btnLogout.addActionListener(e -> logout());

//viewCart.addActionListener(e->cart());

}

private void cart(){

// Cart u1=new Cart();

// u1.setVisible(true);

// setVisible(false);

//

}

private void bookAppointment() throws SQLException, ClassNotFoundException {

String name = (String) cmbVaccineName.getSelectedItem();

//int quantity = Integer.parseInt(txtQuantity.getText());

LoginExample l1=new LoginExample();

int price=0;

for (int i = 0; i < tablemodel1.getRowCount(); i++) {

for (int j = 0; j < tablemodel1.getColumnCount(); j++) {

if (tablemodel1.getValueAt(i, 0).equals(name)) {

price=(Integer)tablemodel1.getValueAt(i, 2);

break;

}

}

if(removeVaccine()){

// try {

// // l1.updateSales(1, 1);

// } catch (SQLException ex) {

// Logger.getLogger(User.class.getName()).log(Level.SEVERE, null, ex);

// }

// txtQuantity.setText("");

}

}

var+="Hospital : CentralHospital\n";

var+="Vaccine : "+name+"\n";

var+="\nTime : "+txtQuantity.getSelectedItem().toString()+"\n";

var+=l1.searchName(Module.userid);

if(var.contains("Second")){

l1.update(Module.userid, 2,name);

var+="\nDose : Second";

}

else{

l1.update(Module.userid, 1,name);

var+="\nDose : First\n";

}

setVisible(false);

PatientSlip.text1+=var;

PatientSlip ps=new PatientSlip();

ps.setVisible(true);

}

private void back(){

PatientPanel u1=new PatientPanel();

u1.setVisible(true);

setVisible(false);

}

private void logout() {

LoginPage manager = new LoginPage();

manager.setVisible(true);

setVisible(false);

}

private boolean removeVaccine() {

String s1=(String) cmbVaccineName.getSelectedItem();

int selectedRow = 0;

for (int i = 0; i < tablemodel1.getRowCount(); i++) {

for (int j = 0; j < tablemodel1.getColumnCount(); j++) {

if (tablemodel1.getValueAt(i, 0).equals(s1)) {

selectedRow = i;

break;

}

}

}

if (selectedRow != -1) {

int quantity=Integer.parseInt(tablemodel1.getValueAt(selectedRow, 2).toString());

String price1=(tablemodel1.getValueAt(selectedRow, 1).toString());

if (quantity<2){

JOptionPane.showMessageDialog(null, "Out of stock");

}

else{

LoginExample l=new LoginExample();

quantity-=1;

try {

l.updateItems(s1, quantity, 2);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

//tableModel.setValueAt(quantity, selectedRow, 2);

Vaccine f1=new Vaccine(s1,price1 ,quantity);

tablemodel1.addVaccine(f1);

tablemodel1.removeVaccine(selectedRow);

return true;

}

}

return false;

}

}

**City.java:**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package covidmanagementsystem;

import java.awt.BorderLayout;

import java.awt.FlowLayout;

import java.sql.SQLException;

import java.util.ArrayList;

import javax.swing.\*;

public class City extends JFrame {

// Declare GUI components

private JLabel lblVaccineName;

public static JComboBox cmbVaccineName;

private JLabel lblQuantity;

private JComboBox txtQuantity;

private JButton btnOrder;

private JButton btnLogout;

private JTable tblOrders;

private JScrollPane scrollPane;

private JButton viewCart;

private JButton btnback;

public String var;

// Declare data model for table

//private OrderTableModel tableModel;

private VaccineTable tablemodel1;

// public String[]it;

public static ArrayList<String> arr;

public City() {

var=new String();

// Set up GUI components

LoginExample l1=new LoginExample();

arr=new ArrayList<>();

String []it={};

try {

it=l1.getVaccines(1);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

lblVaccineName = new JLabel("Vaccine Name:");

cmbVaccineName = new JComboBox(it);

// lblQuantity = new JLabel("Quantity:");

String[]time={"10:00 am","12:00 pm","2:00 pm"};

txtQuantity = new JComboBox(time);

btnOrder = new JButton("Book appointment");

btnLogout = new JButton("Exit");

// viewCart=new JButton("View Cart");

btnback =new JButton("back");

// LoginExample l1=new LoginExample();

// tableModel = new OrderTableModel();

try {

// Set up table and data model

tablemodel1=l1.getItems(1);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

tblOrders = new JTable(tablemodel1);

scrollPane = new JScrollPane(tblOrders);

// Set up layout

setLayout(new BorderLayout());

JPanel pnlInput = new JPanel(new FlowLayout());

pnlInput.add(lblVaccineName);

pnlInput.add(cmbVaccineName);

//pnlInput.add(lblQuantity);

pnlInput.add(txtQuantity);

pnlInput.add(btnOrder);

pnlInput.add(btnLogout);

// pnlInput.add(viewCart);

pnlInput.add(btnback);

add(pnlInput, BorderLayout.NORTH);

add(scrollPane, BorderLayout.CENTER);

// Set up event handlers

btnback.addActionListener(e->back());

btnOrder.addActionListener(e -> {

try {

bookAppointment();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

} catch (ClassNotFoundException ex) {

JOptionPane.showMessageDialog(null, ex);

}

});

btnLogout.addActionListener(e -> logout());

// viewCart.addActionListener(e->cart());

}

private void bookAppointment() throws SQLException, ClassNotFoundException {

String name = (String) cmbVaccineName.getSelectedItem();

//int quantity = Integer.parseInt(txtQuantity.getText());

LoginExample l1=new LoginExample();

int price=0;

for (int i = 0; i < tablemodel1.getRowCount(); i++) {

for (int j = 0; j < tablemodel1.getColumnCount(); j++) {

if (tablemodel1.getValueAt(i, 0).equals(name)) {

price=(Integer)tablemodel1.getValueAt(i, 2);

break;

}

}

if(removeVaccine()){

// try {

// // l1.updateSales(1, 1);

// } catch (SQLException ex) {

// Logger.getLogger(User.class.getName()).log(Level.SEVERE, null, ex);

// }

// txtQuantity.setText("");

}

}

var+="Hospital : CityHospital\n";

var+="Vaccine : "+name+"\n";

var+="\nTime : "+txtQuantity.getSelectedItem().toString()+"\n";

var+=l1.searchName(Module.userid);

if(var.contains("Second")){

l1.update(Module.userid, 2,name);

var+="\nDose : Second\n";

}

else{

var+="\nDose : First\n";

l1.update(Module.userid, 1,name);

}

PatientSlip.text1+=var;

PatientSlip ps=new PatientSlip();

ps.setVisible(true);

setVisible(false);

}

private void back(){

PatientPanel u1=new PatientPanel();

u1.setVisible(true);

setVisible(false);

}

private void logout() {

LoginPage manager = new LoginPage();

manager.setVisible(true);

setVisible(false);

}

private boolean removeVaccine() {

String s1=(String) cmbVaccineName.getSelectedItem();

int selectedRow = 0;

for (int i = 0; i < tablemodel1.getRowCount(); i++) {

for (int j = 0; j < tablemodel1.getColumnCount(); j++) {

if (tablemodel1.getValueAt(i, 0).equals(s1)) {

selectedRow = i;

break;

}

}

}

if (selectedRow != -1) {

int quantity=Integer.parseInt(tablemodel1.getValueAt(selectedRow, 2).toString());

String price1=(tablemodel1.getValueAt(selectedRow, 1).toString());

if (quantity<2){

JOptionPane.showMessageDialog(null, "Out of stock");

}

else{

LoginExample l=new LoginExample();

quantity-=1;

try {

l.updateItems(s1, quantity, 1);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, ex);

}

Vaccine f1=new Vaccine(s1,price1 ,quantity);

tablemodel1.addVaccine(f1);

tablemodel1.removeVaccine(selectedRow);

return true;

}

}

return false;

}

}

**Covid19Vaccinemanagement.JAVA:**

package covidmanagementsystem;

public class Covid19Vaccinemanagement {

public static void main(String[] args) {

Welcome w=new Welcome();

w.setVisible(true);

}

}

**Login Example.JAVA:**

package covidmanagementsystem;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.\*;

import javax.swing.JOptionPane;

public class LoginExample {

public static int id = -1;

Connection conn = null;

Statement stmt = null;

PreparedStatement pst = null;

ResultSet rs = null;

String name, pwd = null;

public LoginExample() {

try {

Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");

conn = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\HP\\OneDrive\\Desktop\\Database3.accdb");

JOptionPane.showMessageDialog(null, "Succesfull database Connection");

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex);

}

}

public void update(int id, int choice, String Vaccine) throws SQLException, ClassNotFoundException {

if (choice == 1) {

String var = "Done";

String sql = "UPDATE Patient SET FirstDose='" + var + "',Vaccine='" + Vaccine + "' WHERE ID='" + id + "' ";

pst = conn.prepareStatement(sql);

int rs = pst.executeUpdate();

if (rs > 0) {

JOptionPane.showMessageDialog(null, "Successfully Updated");

}

} else if (choice == 2) {

String var = "Done";

String sql = "UPDATE Patient SET SecondDose ='" + var + "' WHERE ID='" + id + "'";

pst = conn.prepareStatement(sql);

int rs = pst.executeUpdate();

if (rs > 0) {

JOptionPane.showMessageDialog(null, "Successfully Updated");

}

}

}

public void addItems(String name, String man, int quantity, int choice) {

if (choice == 1) {

String sql = "insert into vaccine(Vaccine,Manufacturer,Quantity) values ('" + name + "','" + man + "','" + quantity + "')";

try {

stmt = conn.createStatement();

int res = stmt.executeUpdate(sql);

if (res > 0) {

JOptionPane.showMessageDialog(null, "Inserted");

} else {

JOptionPane.showMessageDialog(null, "Error");

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex);

}

}

if (choice == 2) {

String sql = "insert into vaccine2(Vaccine,Manufacturer,Quantity) values ('" + name + "','" + man + "','" + quantity + "')";

try {

stmt = conn.createStatement();

int res = stmt.executeUpdate(sql);

if (res > 0) {

JOptionPane.showMessageDialog(null, "Inserted");

} else {

JOptionPane.showMessageDialog(null, "Error");

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex);

}

}

}

public void adduser(String uname, String pass) {

String sql = "insert into Patient(PatientName,Cnic) values ('" + uname + "','" + pass + "')";

try {

stmt = conn.createStatement();

int res = stmt.executeUpdate(sql);

if (res > 0) {

JOptionPane.showMessageDialog(null, "Inserted");

} else {

JOptionPane.showMessageDialog(null, "Error");

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex);

}

}

public VaccineTable getItems(int choice) throws SQLException {

String name = new String();

VaccineTable f1 = new VaccineTable();

int Quantitiy;

String Manufacturer;

String var = new String();

if (choice == 1) {

String sql = "Select Vaccine,Manufacturer,Quantity from vaccine ";

int counter = 0;

pst = conn.prepareStatement(sql);

rs = pst.executeQuery();

while (rs.next()) {

name = rs.getString("Vaccine");

Manufacturer = (rs.getString("Manufacturer"));

Quantitiy = Integer.parseInt(rs.getString("Quantity"));

Vaccine f = new Vaccine(name, Manufacturer, Quantitiy);

f1.addVaccine(f);

//pwd= rs.getString("Password");

}

if (name != null) {

JOptionPane.showMessageDialog(null, "Search Successfull");

} else {

JOptionPane.showMessageDialog(null, "ERROR");

}

} else {

String sql = "Select Vaccine,Manufacturer,Quantity from vaccine1 ";

int counter = 0;

pst = conn.prepareStatement(sql);

rs = pst.executeQuery();

while (rs.next()) {

name = rs.getString("Vaccine");

Manufacturer = (rs.getString("Manufacturer"));

Quantitiy = Integer.parseInt(rs.getString("Quantity"));

Vaccine f = new Vaccine(name, Manufacturer, Quantitiy);

f1.addVaccine(f);

}

if (name != null) {

JOptionPane.showMessageDialog(null, "Search Successful");

} else {

JOptionPane.showMessageDialog(null, "ERROR");

}

}

return f1;

}

public int search(String uname, String pass) throws SQLException {

name = null;

pwd = null;

String sql = "Select ID from Login where Username ='" + uname + "' and Password ='" + pass + "' ";

boolean b;

pst = conn.prepareStatement(sql);

rs = pst.executeQuery();

while (rs.next()) {

name = rs.getString("ID");

}

if (name != null) {

JOptionPane.showMessageDialog(null, "Search Successful");

return Integer.parseInt(name);

} else {

JOptionPane.showMessageDialog(null, "ERROR");

}

return -1;

}

void delete(int id) throws SQLException, ClassNotFoundException {

String sql = "DELETE FROM Login WHERE (ID) = ('" + id + "')";

pst = conn.prepareStatement(sql);

int rs = pst.executeUpdate();

}

public void updateItems(String name, int quantity, int choice) throws SQLException {

if (choice == 1) {

String sql = "UPDATE vaccine SET Quantity='" + quantity + "' WHERE Vaccine='" + name + "'";

pst = conn.prepareStatement(sql);

int rs = pst.executeUpdate();

if (rs > 0) {

JOptionPane.showMessageDialog(null, "Successfully Updated");

}

}

if (choice == 2) {

String sql = "UPDATE vaccine1 SET Quantity='" + quantity + "' WHERE Vaccine='" + name + "'";

pst = conn.prepareStatement(sql);

int rs = pst.executeUpdate();

if (rs > 0) {

JOptionPane.showMessageDialog(null, "Successfully Updated");

}

}

}

public String searchName(int id) throws SQLException {

String sql = "Select Patientname,FirstDose,SecondDose from Patient where ID='" + id + "' ";

String var = new String();

int counter = 0;

String var1 = new String();

String var2 = new String();

pst = conn.prepareStatement(sql);

rs = pst.executeQuery();

while (rs.next()) {

var += "Patient name : " + (rs.getString("PatientName"));

var1 = rs.getString("FirstDose");

var2 = rs.getString("SecondDose");

}

if (var1 != null) {

if (var2 != null) {

var += "Both doses done";

} else {

var += "FirstDose : Done \nSecond Dose : Done ";

}

} else {

var += "FirstDose : Done ";

}

return var;

}

public int search(String cnic) throws SQLException {

// String Cnic=new String();

VaccineTable f1 = new VaccineTable();

String sql = "Select ID from Patient where Cnic='" + cnic + "' ";

int counter = 0;

pst = conn.prepareStatement(sql);

rs = pst.executeQuery();

while (rs.next()) {

id = Integer.parseInt(rs.getString("ID"));

}

return id;

}

public String[] getVaccines(int choice) throws SQLException {

String[] arr = new String[20];

if (choice == 1) {

String sql = "Select Vaccine from vaccine ";

int counter = 0;

pst = conn.prepareStatement(sql);

rs = pst.executeQuery();

while (rs.next()) {

arr[counter] = rs.getString("Vaccine");

counter++;

}

} else {

String sql = "Select Vaccine from vaccine1 ";

int counter = 0;

pst = conn.prepareStatement(sql);

rs = pst.executeQuery();

while (rs.next()) {

arr[counter] = rs.getString("Vaccine");

counter++;

}

}

return arr;

}

}

**Vaccine.JAVA:**

package covidmanagementsystem;

public class Vaccine {

private String name;

private String manufacturer;

private int quantity;

public Vaccine(String name, String manufacturer, int quantity) {

this.name = name;

this.manufacturer = manufacturer;

this.quantity = quantity;

}

public String getName() {

return name;

}

public String getManufacturer() {

return manufacturer;

}

public int getQuantity() {

return quantity;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

}

**VaccineTable.java:**

package covidmanagementsystem;

import javax.swing.table.AbstractTableModel;

import java.util.List;

import java.util.ArrayList;

public class VaccineTable extends AbstractTableModel {

// Declare column names and data

private String[] columnNames = {"Vaccine", "Manufacturer","Quantitiy",""};

private List<Vaccine> data = new ArrayList<>();

@Override

public int getRowCount() {

return data.size();

}

@Override

public int getColumnCount() {

return columnNames.length;

}

@Override

public String getColumnName(int column) {

return columnNames[column];

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

Vaccine v = data.get(rowIndex);

if (columnIndex == 0) {

return v.getName();

} else if (columnIndex == 1) {

return v.getManufacturer();

}

else if (columnIndex==2){

return v.getQuantity();

}else {

return null;

}

}

public void addVaccine(Vaccine v) {

data.add(v);

fireTableRowsInserted(data.size() - 1, data.size() - 1);

}

public void removeVaccine(int rowIndex) {

data.remove(rowIndex);

fireTableRowsDeleted(rowIndex, rowIndex);

}

}

**ADMIN ACCESS PANEL.JAVA:**

package covidmanagementsystem;

public class AdminAcessPanel extends javax.swing.JFrame {

/\*\*

\* Creates new form AdminAcessPanel

\*/

public AdminAcessPanel() {

initComponents();

}

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

// TODO add your handling code here:

AdminCentral a=new AdminCentral();

a.setVisible(true);

setVisible(false);

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:'

setVisible(true);

LoginPage l1=new LoginPage();

l1.setVisible(true);

dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

setVisible(false);

AdminCity a=new AdminCity();

a.setVisible(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(AdminAcessPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(AdminAcessPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(AdminAcessPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(AdminAcessPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new AdminAcessPanel().setVisible(true);

}

});

}

**LOGIN PAGE.JAVA:**

package covidmanagementsystem;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class LoginPage extends javax.swing.JFrame {

/\*\*

\* Creates new form LoginPage

\*/

public LoginPage() {

initComponents();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String username=un.getText();

String password=pass.getText();

LoginExample l1=new LoginExample();

int id=0;

try {

id=l1.search(username, password);

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "error");

}

if(id==1){

// MenuPage obj=new MenuPage();

// obj.setVisible(true);

// dispose();

AdminAcessPanel a=new AdminAcessPanel();

setVisible(false);

a.setVisible(true);

}

// else if(id>1){

// MenuPage m=new MenuPage();

// m.setVisible(true);

// setVisible(false);

//}

else{

JOptionPane.showMessageDialog(rootPane, "Username or password is incorrect");

}

}

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

Welcome w=new Welcome();

w.setVisible(true);

setVisible(false);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LoginPage.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new LoginPage().setVisible(true);

}

});

}

**MODULE.JAVA:**

package covidmanagementsystem;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.Timer;

/\*\*

\*

\* @author Muhammad Taha Azam

\*/

public class Module extends javax.swing.JFrame {

public static int userid=0;

public static boolean v1=false;

/\*\*

\* Creates new form Module1

\*/

public Module() {

initComponents();

showDate();

showTime();

}

void showDate(){

Date d=new Date();

SimpleDateFormat s=new SimpleDateFormat("yyyy-MM-dd");

datelab.setText(s.format(d));

}

void showTime(){

new Timer(0,new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

Date d=new Date();

SimpleDateFormat s=new SimpleDateFormat("hh:mm:ss a");

timelab.setText(s.format(d));

// throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

}).start();

}

private void jLabel8MouseClicked(java.awt.event.MouseEvent evt) {

Welcome obj=new Welcome();

obj.setVisible(true);

// dispose();

// TODO add your handling code here:

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String cnic=CNIC.getText();

String name=pn.getText();

if (cnic.isEmpty()||name.isEmpty()){

JOptionPane.showMessageDialog(null,"enter full details please");

}

else{

try {

// TODO add your handling code here:

LoginExample l1=new LoginExample();

if(l1.search(cnic)<=0){

l1.adduser(name, cnic);

userid=l1.search(cnic);

}

else{

userid=l1.search(cnic);

JOptionPane.showMessageDialog(null, "Patient "+name+" exists" );

v1=true;

}

} catch (SQLException ex) {

Logger.getLogger(Module.class.getName()).log(Level.SEVERE, null, ex);

}

}

PatientPanel p1=new PatientPanel();

p1.setVisible(true);

setVisible(false);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Module.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Module.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Module.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Module.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new Module().setVisible(true);

}

});

}

**PATIENT PANEL.JAVA:**

package covidmanagementsystem;

public class PatientPanel extends javax.swing.JFrame {

public PatientPanel() {

initComponents();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

setVisible(false);

City c=new City();

c.setVisible(true);

// dispose();

}

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

setVisible(false);

Central c=new Central();

c.setVisible(true);

//dispose();

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(PatientPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(PatientPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(PatientPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(PatientPanel.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new PatientPanel().setVisible(true);

}

});

}

**PATIENT SLIP.JAVA:**

package covidmanagementsystem;

/\*\*

\*

\* @author noorm

\*/

public class PatientSlip extends javax.swing.JFrame {

/\*\*

\* Creates new form PatientSlip

\*/

public static String text1=new String();

public PatientSlip() {

initComponents();

jTextArea1.setText(text1);

}

package covidmanagementsystem;

/\*\*

\*

\* @author noorm

\*/

public class PatientSlip extends javax.swing.JFrame {

/\*\*

\* Creates new form PatientSlip

\*/

public static String text1=new String();

public PatientSlip() {

initComponents();

jTextArea1.setText(text1);

}

**WELCOME.JAVA:**

package covidmanagementsystem;

/\*\*

\*

\* @author noorm

\*/

public class Welcome extends javax.swing.JFrame {

/\*\*

\* Creates new form Welcome

\*/

public Welcome() {

initComponents();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Module m = new Module();

m.setVisible(true);

setVisible(false);

dispose();

}

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

// TODO add your handling code here:

LoginPage m = new LoginPage();

m.setVisible(true);

setVisible(false);

dispose();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Welcome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new Welcome().setVisible(true);

}

});

}

1. **CONCLUSION**

COVID Vaccine Management system aims at developing a fully functional computerized system to maintain all the day-to-day activity of a COVID Vaccine department. Through this system, health care providers would be able to:

* Registration for COVID-19 vaccine.
* Choose between City Hospital and Central Hospital.
* Let recipients self-schedule their appointments.
* Let admin update user vaccine dose.

Let admin keep track of vaccination and order whenever required.