BUS RESERVATION SYSTEM

DEPT.NO : 19/UCSA/209

SUBJ.CODE : 19CS/ME/AJ45

OBJECTIVE :

The main objection of the Project on Bus Ticket Booking System is to manage the details of Bus ,Tickets, Booking, Customer, Seats. It manages all the information about Bus, Seats, Bus. The purpose of the project is to build an application program to reduce the manual work for managing the Bus, Tickets, Agent ,Booking. It tracks all the details about the Booking, Customer, Seats.

**The modules of the project are listed as below:**

* Admin Login
* User login/registration
* Seat availability check
* Seat booking
* Additional facilities
* Seat cancellation

SOURCE CODE :

Code.java

package Bus\_Ticket\_Reservation;

import java.sql.\*;

import java.util.Optional;

import java.time.format.DateTimeFormatter;

import javafx.application.Application;

import javafx.beans.property.SimpleStringProperty;

import javafx.beans.value.ChangeListener;

import javafx.beans.value.ObservableValue;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.geometry.VPos;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.control.TableColumn.CellDataFeatures;

import javafx.scene.control.TableView.TableViewSelectionModel;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.layout.\*;

import javafx.scene.paint.Color;

import javafx.scene.text.Font;

import javafx.scene.text.Text;

import javafx.stage.Modality;

import javafx.stage.Stage;

import javafx.stage.StageStyle;

import javafx.util.Callback;

public class Code extends Application {

public static void main(String args[])

{

launch(args);

}

@Override

public void start(Stage primaryStage) {

HBox mainpage = new HBox();

Image img = new Image(getClass().getResourceAsStream("MainImage1.jpg"));

ImageView imgv = new ImageView(img);

imgv.setFitWidth(950);

imgv.setFitHeight(1010);

VBox login = new VBox();

Label logint = new Label("LOGIN");

logint.setFont(new Font("Arial",36));

FlowPane lfp1 = new FlowPane(logint);

lfp1.setAlignment(Pos.CENTER);

lfp1.setPadding(new Insets(30));

GridPane logingp = new GridPane();

Image imgui = new Image(getClass().getResourceAsStream("usericon.png"));

ImageView imguiv = new ImageView(imgui);

imguiv.setFitWidth(50);

imguiv.setFitHeight(50);

TextField usernametf = new TextField();

usernametf.setPrefSize(270, 50);

usernametf.setPromptText("Username");

Image imgpi = new Image(getClass().getResourceAsStream("pwdicon.png"));

ImageView imgpiv = new ImageView(imgpi);

imgpiv.setFitWidth(50);

imgpiv.setFitHeight(50);

PasswordField pwdtf = new PasswordField();

pwdtf.setPrefSize(270, 50);

pwdtf.setPromptText("Password");

logingp.setHgap(30);

logingp.setVgap(30);

logingp.add(imguiv, 0, 0);

logingp.add(usernametf, 1, 0);

logingp.add(imgpiv, 0, 1);

logingp.add(pwdtf, 1, 1);

Button loginbtn = new Button("Login");

loginbtn.setId("loginbtn");

Alert a = new Alert(AlertType.NONE);

if(usernametf.getText().isEmpty() && pwdtf.getText().isEmpty()){

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Fields are empty");

a.show();

}

else if(usernametf.getText().isEmpty()){

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Username Field is Empty");

a.show();

}

else{

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Password Field is Empty");

a.show();

}

loginbtn.setOnAction(e -> {

try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from login where login.username='"+usernametf.getText()+"' and login.password='"+pwdtf.getText()+"'");

if(res.next()){

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Successful");

Optional<ButtonType> result = a.showAndWait();

ButtonType button = result.orElse(ButtonType.CANCEL);

if (button == ButtonType.OK) {

primaryStage.close();

login();

}

}

else{

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Failed");

a.showAndWait();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

});

loginbtn.setPrefSize(190, 50);

FlowPane lfp2 = new FlowPane(loginbtn);

lfp2.setAlignment(Pos.CENTER);

lfp2.setPadding(new Insets(40,0,10,0));

Button newloginbtn = new Button("Sign Up");

newloginbtn.setId("newloginbtn");

newloginbtn.setOnAction(e -> {

signup();

});

newloginbtn.setPrefSize(190, 50);

FlowPane lfp3 = new FlowPane(newloginbtn);

lfp3.setAlignment(Pos.CENTER);

lfp3.setPadding(new Insets(10,0,10,0));

Button adminloginbtn = new Button("Admin Login");

adminloginbtn.setId("adminloginbtn");

adminloginbtn.setOnAction(e -> {

adminlogin();

});

adminloginbtn.setPrefSize(190, 50);

FlowPane lfp4 = new FlowPane(adminloginbtn);

lfp4.setAlignment(Pos.CENTER);

lfp4.setPadding(new Insets(10,0,10,0));

login.getChildren().addAll(lfp1,logingp,lfp2,lfp3,lfp4);

login.setPadding(new Insets(280));

mainpage.getChildren().addAll(imgv,login);

Scene = new Scene(mainpage);

scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

primaryStage.setTitle("Paradise Travel");

primaryStage.setScene(scene);

primaryStage.show();

}

public void signup(){

Alert a = new Alert(AlertType.NONE);

final Stage dialog=new Stage();

dialog.initModality(Modality.APPLICATION\_MODAL);

VBox signup = new VBox();

Label signupt = new Label("SIGN UP");

signupt.setFont(new Font("Arial",36));

FlowPane sufp1 = new FlowPane(signupt);

sufp1.setAlignment(Pos.CENTER);

sufp1.setPadding(new Insets(30));

GridPane signupgp = new GridPane();

TextField firstnametf = new TextField();

firstnametf.setPrefSize(270, 50);

firstnametf.setPromptText("First Name");

TextField lastnametf = new TextField();

lastnametf.setPrefSize(270, 50);

lastnametf.setPromptText("Last Name");

TextField usernametf = new TextField();

usernametf.setPrefSize(270, 50);

usernametf.setPromptText("Username");

PasswordField pwdtf = new PasswordField();

pwdtf.setPrefSize(270, 50);

pwdtf.setPromptText("Password");

PasswordField cpwdtf = new PasswordField();

cpwdtf.setPrefSize(270, 50);

cpwdtf.setPromptText("Confirm Password");

Button signupbtn = new Button("Sign Up");

signupbtn.setPrefSize(190, 50);

signupbtn.setId("signupbtn");

Button cancelbtn = new Button("Cancel");

cancelbtn.setPrefSize(190, 50);

cancelbtn.setId("cancelbtn");

signupbtn.setOnAction(e -> {

try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("insert into LOGIN values(?,?,?,?)");

ps.setString(1,firstnametf.getText());

ps.setString(2,lastnametf.getText());

ps.setString(3,usernametf.getText());

ps.setString(4,pwdtf.getText());

ps.executeUpdate();

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Successful");

a.show();

login();

});

cancelbtn.setOnAction(e -> {

dialog.close();

});

signupgp.setHgap(30);

signupgp.setVgap(30);

signupgp.add(firstnametf, 0, 0);

signupgp.add(lastnametf, 1, 0);

signupgp.add(usernametf, 0, 1);

signupgp.add(pwdtf, 0, 2);

signupgp.add(cpwdtf, 0, 3);

signupgp.add(signupbtn, 0, 4);

signupgp.add(cancelbtn, 1, 4);

signup.getChildren().addAll(sufp1,signupgp);

signup.setPadding(new Insets(100));

Scene scene2=new Scene(signup);

scene2.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog.setScene(scene2);

dialog.initStyle(StageStyle.DECORATED);

dialog.showAndWait();

}

public void adminlogin(){

final Stage dialog=new Stage();

dialog.initModality(Modality.APPLICATION\_MODAL);

HBox mainpage = new HBox();

Image img = new Image(getClass().getResourceAsStream("MainImage1.jpg"));

ImageView imgv = new ImageView(img);

imgv.setFitWidth(950);

imgv.setFitHeight(1000);

VBox login = new VBox();

Label logint = new Label("ADMIN LOGIN");

logint.setFont(new Font("Arial",40));

FlowPane lfp1 = new FlowPane(logint);

lfp1.setAlignment(Pos.CENTER);

lfp1.setPadding(new Insets(30));

GridPane logingp = new GridPane();

Image imgui = new Image(getClass().getResourceAsStream("usericon.png"));

ImageView imguiv = new ImageView(imgui);

imguiv.setFitWidth(50);

imguiv.setFitHeight(50);

TextField usernametf = new TextField();

usernametf.setPrefSize(270, 50);

usernametf.setPromptText("Username");

Image imgpi = new Image(getClass().getResourceAsStream("pwdicon.png"));

ImageView imgpiv = new ImageView(imgpi);

imgpiv.setFitWidth(50);

imgpiv.setFitHeight(50);

PasswordField pwdtf = new PasswordField();

pwdtf.setPrefSize(270, 50);

pwdtf.setPromptText("Password");

logingp.setHgap(30);

logingp.setVgap(30);

logingp.add(imguiv, 0, 0);

logingp.add(usernametf, 1, 0);

logingp.add(imgpiv, 0, 1);

logingp.add(pwdtf, 1, 1);

Button loginbtn = new Button("Login");

loginbtn.setId("loginbtn");

Alert a = new Alert(AlertType.NONE);

loginbtn.setOnAction(e -> {

try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from adminlogin where adminlogin.username='"+usernametf.getText()+"' and adminlogin.password='"+pwdtf.getText()+"'");

if(res.next()){

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Successful");

Optional<ButtonType> result = a.showAndWait();

ButtonType button = result.orElse(ButtonType.CANCEL);

if (button == ButtonType.OK) {

addseat();

dialog.close();

login();

}

}

else{

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Failed");

a.showAndWait();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

});

loginbtn.setPrefSize(190, 50);

FlowPane lfp2 = new FlowPane(loginbtn);

lfp2.setAlignment(Pos.CENTER);

lfp2.setPadding(new Insets(40,0,10,0));

login.getChildren().addAll(lfp1,logingp,lfp2);

login.setPadding(new Insets(280));

mainpage.getChildren().addAll(imgv,login);

Scene scene3=new Scene(mainpage);

scene3.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog.setScene(scene3);

dialog.initStyle(StageStyle.DECORATED);

dialog.showAndWait();

}

public void addseat(){

Alert a = new Alert(AlertType.NONE);

final Stage dialog=new Stage();

dialog.initModality(Modality.APPLICATION\_MODAL);

VBox addseat = new VBox();

Label addseatl = new Label("ADD SEAT");

addseatl.setFont(new Font("Arial",36));

FlowPane asfp = new FlowPane(addseatl);

asfp.setAlignment(Pos.CENTER);

asfp.setPadding(new Insets(30));

GridPane addseatgp = new GridPane();

Label busno = new Label("Bus No ");

ComboBox busnocb = new ComboBox();

busnocb.getItems().addAll("101","102","103","104","105","106","107","108","109","110","111","112","113","114");

busnocb.setPrefSize(270, 50);

Label datel = new Label("Date ");

DatePicker dp = new DatePicker();

Button addbtn = new Button("Add");

addbtn.setOnAction(e -> {

String busnumber =busnocb.getValue().toString();

String date = dp.getValue().format(DateTimeFormatter.ofPattern("yyyy-MM-dd"));

for(int i=1;i<=20;i++){

try{

if(i%2!=0){

String seats=i+"W";

String status="Unbooked";

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("insert into SEAT(busid,seats,date,status)values(?,?,?,?)");

ps.setString(1,busnumber);

ps.setString (2,seats);

ps.setString(3,date);

ps.setString(4,status);

ps.executeUpdate();

}

else{

String seats=i+"A";

String status="Unbooked";

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("insert into SEAT(busid,seats,date,status)values(?,?,?,?)");

ps.setString(1,busnumber);

ps.setString (2,seats);

ps.setString(3,date);

ps.setString(4,status);

ps.executeUpdate();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

}

for(int i=21;i<=40;i++){

try{

if(i%2!=0){

String seats=i+"W";

String status="Unbooked";

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("insert into SEAT(busid,seats,date,status)values(?,?,?,?)");

ps.setString(1,busnumber);

ps.setString (2,seats);

ps.setString(3,date);

ps.setString(4,status);

ps.executeUpdate();

}

else{

String seats=i+"A";

String status="Unbooked";

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("insert into SEAT(busid,seats,date,status)values(?,?,?,?)");

ps.setString(1,busnumber);

ps.setString (2,seats);

ps.setString(3,date);

ps.setString(4,status);

ps.executeUpdate();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

}

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Sheet Added");

Optional<ButtonType> result = a.showAndWait();

ButtonType button = result.orElse(ButtonType.CANCEL);

if (button == ButtonType.OK) {

addseat();

}

});

addbtn.setId("asaddbtn");

addbtn.setPrefSize(190, 50);

Button cbtn = new Button("Cancel");

cbtn.setOnAction(e -> {

dialog.close();

mainpage();

});

cbtn.setId("ascbtn");

cbtn.setPrefSize(190, 50);

addseatgp.setHgap(30);

addseatgp.setVgap(30);

addseatgp.add(busno,0,0);

addseatgp.add(busnocb,1,0);

addseatgp.add(datel,0,1);

addseatgp.add(dp,1,1);

addseatgp.add(addbtn,0,2);

addseatgp.add(cbtn,1,2);

addseat.getChildren().addAll(asfp,addseatgp);

addseat.setPadding(new Insets(100));

Scene scene4=new Scene(addseat);

scene4.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog.setScene(scene4);

dialog.initStyle(StageStyle.DECORATED);

dialog.showAndWait();

}

public void mainpage(){

final Stage dialog=new Stage();

dialog.initModality(Modality.APPLICATION\_MODAL);

HBox mainpage = new HBox();

Image img = new Image(getClass().getResourceAsStream("MainImage1.jpg"));

ImageView imgv = new ImageView(img);

imgv.setFitWidth(950);

imgv.setFitHeight(1000);

VBox login = new VBox();

Label logint = new Label("LOGIN");

logint.setFont(new Font("Arial",36));

FlowPane lfp1 = new FlowPane(logint);

lfp1.setAlignment(Pos.CENTER);

lfp1.setPadding(new Insets(30));

GridPane logingp = new GridPane();

Image imgui = new Image(getClass().getResourceAsStream("usericon.png"));

ImageView imguiv = new ImageView(imgui);

imguiv.setFitWidth(50);

imguiv.setFitHeight(50);

TextField usernametf = new TextField();

usernametf.setPrefSize(270, 50);

usernametf.setPromptText("Username");

Image imgpi = new Image(getClass().getResourceAsStream("pwdicon.png"));

ImageView imgpiv = new ImageView(imgpi);

imgpiv.setFitWidth(50);

imgpiv.setFitHeight(50);

PasswordField pwdtf = new PasswordField();

pwdtf.setPrefSize(270, 50);

pwdtf.setPromptText("Password");

logingp.setHgap(30);

logingp.setVgap(30);

logingp.add(imguiv, 0, 0);

logingp.add(usernametf, 1, 0);

logingp.add(imgpiv, 0, 1);

logingp.add(pwdtf, 1, 1);

Button loginbtn = new Button("Login");

loginbtn.setId("loginbtn");

Alert a = new Alert(AlertType.NONE);

loginbtn.setOnAction(e -> {

try

{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from login where login.username='"+usernametf.getText()+"' and login.password='"+pwdtf.getText()+"'");

if(res.next()){

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Successful");

Optional<ButtonType> result = a.showAndWait();

ButtonType button = result.orElse(ButtonType.CANCEL);

if (button == ButtonType.OK) {

dialog.close();

login();

}

}

else{

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Failed");

a.showAndWait();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

});

loginbtn.setPrefSize(190, 50);

FlowPane lfp2 = new FlowPane(loginbtn);

lfp2.setAlignment(Pos.CENTER);

lfp2.setPadding(new Insets(40,0,10,0));

Button newloginbtn = new Button("Sign Up");

newloginbtn.setId("newloginbtn");

newloginbtn.setOnAction(e -> {

signup();

});

newloginbtn.setPrefSize(190, 50);

FlowPane lfp3 = new FlowPane(newloginbtn);

lfp3.setAlignment(Pos.CENTER);

lfp3.setPadding(new Insets(10,0,10,0));

Button adminloginbtn = new Button("Admin Login");

adminloginbtn.setId("adminloginbtn");

adminloginbtn.setOnAction(e -> {

adminlogin();

});

adminloginbtn.setPrefSize(190, 50);

FlowPane lfp4 = new FlowPane(adminloginbtn);

lfp4.setAlignment(Pos.CENTER);

lfp4.setPadding(new Insets(10,0,10,0));

login.getChildren().addAll(lfp1,logingp,lfp2,lfp3,lfp4);

login.setPadding(new Insets(280));

mainpage.getChildren().addAll(imgv,login);

Scene scene5=new Scene(mainpage);

scene5.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog.setScene(scene5);

dialog.initStyle(StageStyle.DECORATED);

dialog.showAndWait();

}

public ObservableList<ObservableList> data;

public TableView;

public void login(){

final Stage dialog=new Stage();

dialog.initModality(Modality.APPLICATION\_MODAL);

Alert a = new Alert(AlertType.NONE);

BorderPane bpane = new BorderPane();

GridPane headgp = new GridPane();

Image imglogo = new Image(getClass().getResourceAsStream("logo.png"));

ImageView imglogov = new ImageView(imglogo);

imglogov.setFitWidth(80);

imglogov.setFitHeight(100);

Label bustick = new Label("BUS TICKETS");

bustick.setId("head");

Label help = new Label("Help");

help.setId("head");

Label cantick = new Label("Cancel Ticket");

cantick.setOnMouseClicked(e -> {

canceltick();

});

cantick.setId("head");

headgp.setHgap(10);

headgp.add(imglogov, 20, 0);

headgp.add(bustick, 22, 0);

headgp.add(help, 100, 0);

headgp.add(cantick, 102, 0);

headgp.setId("header");

Image imgmain = new Image(getClass().getResourceAsStream("mainpic.png"));

ImageView imgmainv = new ImageView(imgmain);

imgmainv.setFitWidth(1920);

imgmainv.setFitHeight(650);

FlowPane contentfp = new FlowPane();

ComboBox fromcb = new ComboBox();

fromcb.getItems().addAll(" Chennai"," Bengaluru"," Coimbatore"," Tichy"," Tirunelveli"," Madurai");

fromcb.setPrefSize(270, 50);

fromcb.setId("fromlocicon");

fromcb.setPromptText(" FROM");

ComboBox tocb = new ComboBox();

tocb.getItems().addAll(" Chennai"," Bengaluru"," Coimbatore"," Tichy"," Tirunelveli"," Madurai");

tocb.setPrefSize(270, 50);

tocb.setId("tolocicon");

tocb.setPromptText(" TO");

DatePicker datedp = new DatePicker();

datedp.setPrefSize(220, 50);

datedp.setId("dateicon");

datedp.setPromptText("DATE");

ComboBox bustypecb = new ComboBox();

bustypecb.getItems().addAll(" Seater NONAC"," Seater AC"," Sleeper NONAC"," Sleeper AC");

bustypecb.setPrefSize(270, 50);

bustypecb.setPromptText(" BUS TYPE");

Button btn = new Button("Search Buses");

btn.setOnAction(e -> {

final Stage dialog1=new Stage();

dialog1.initModality(Modality.APPLICATION\_MODAL);

VBox = new VBox();

GridPane headgp1 = new GridPane();

headgp1.setId("header");

Image imglogo1 = new Image(getClass().getResourceAsStream("logo.png"));

ImageView imglogov1 = new ImageView(imglogo1);

imglogov1.setFitWidth(80);

imglogov1.setFitHeight(100);

Label bustick1 = new Label("BUS TICKETS");

bustick1.setId("head");

headgp1.setHgap(10);

headgp1.add(imglogov1, 20, 0);

headgp1.add(bustick1, 22, 0);

if ((fromcb.getValue().equals(" Chennai")) && (tocb.getValue().equals(" Bengaluru"))) {

Label avaserl = new Label("Available Services");

avaserl.setPadding(new Insets(10));

FlowPane avaserfp = new FlowPane(avaserl);

avaserfp.setAlignment(Pos.CENTER);

Label fromtol = new Label(fromcb.getValue()+" to"+tocb.getValue()+" - Journey date "+datedp.getValue());

fromtol.setAlignment(Pos.CENTER\_LEFT);

GridPane ctobgp = new GridPane();

final Separator sepHor = new Separator();

sepHor.setValignment(VPos.CENTER);

GridPane.setConstraints(sepHor, 0, 1);

GridPane.setColumnSpan(sepHor, 11);

ctobgp.getChildren().add(sepHor);

Label busidl = new Label("Bus Id");

Label clofserl = new Label("Class of Service");

Label viarl = new Label("Via Route");

Label depl = new Label("Departure");

Label arrl = new Label("Arrival");

Label orgl = new Label("Origin");

Label desl = new Label("Destination");

Label ratl = new Label("Rating");

Label farel = new Label("Fare");

Label selserl = new Label("Select Service");

Label busidl1 = new Label("101");

Label clofserl1 = new Label("Seater NONAC");

Label viarl1 = new Label("Oosur");

Label depl1 = new Label("08:00 ");

Label arrl1 = new Label("14:00");

Label orgl1 = new Label("Chennai");

Label desl1 = new Label("Bengaluru");

Image imgrat = new Image(getClass().getResourceAsStream("rat1.png"));

ImageView imgratv = new ImageView(imgrat);

imgratv.setFitWidth(50);

imgratv.setFitHeight(43);

Label farel1 = new Label("500");

Button ctobbtn = new Button("Book");

String date = datedp.getValue().format(DateTimeFormatter.ofPattern("yyyy-MM-dd"));

ctobbtn.setOnAction(e1->{

dialog1.close();

booktickets();

});

ctobgp.setHgap(20);

ctobgp.setVgap(20);

ctobgp.add(busidl, 0, 0);

ctobgp.add(clofserl, 1, 0);

ctobgp.add(viarl, 2, 0);

ctobgp.add(depl, 3, 0);

ctobgp.add(arrl, 4, 0);

ctobgp.add(orgl, 5, 0);

ctobgp.add(desl, 6, 0);

ctobgp.add(ratl, 7, 0);

ctobgp.add(farel, 8, 0);

ctobgp.add(selserl, 9, 0);

ctobgp.add(busidl1, 0, 2);

ctobgp.add(clofserl1, 1, 2);

ctobgp.add(viarl1, 2, 2);

ctobgp.add(depl1, 3, 2);

ctobgp.add(arrl1, 4, 2);

ctobgp.add(orgl1, 5, 2);

ctobgp.add(desl1, 6, 2);

ctobgp.add(imgratv, 7, 2);

ctobgp.add(farel1, 8, 2);

ctobgp.add(ctobbtn, 9, 2);

ctobgp.setAlignment(Pos.CENTER);

Button gobackbtn = new Button("Go Back");

gobackbtn.setOnAction(e1->{

dialog1.close();

});

ctobgp.setBorder(new Border(new BorderStroke(Color.valueOf("#d60909"),BorderStrokeStyle.SOLID,CornerRadii.EMPTY,BorderWidths.DEFAULT)));

ctobgp.setPadding(new Insets(50));

vbox.getChildren().addAll(headgp1,avaserl,fromtol,ctobgp,gobackbtn);

Scene scene6=new Scene(vbox);

scene6.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog1.setScene(scene6);

dialog1.initStyle(StageStyle.DECORATED);

dialog1.showAndWait();

}

else{

Label avaserl = new Label("Sorry No Services Available ");

avaserl.setAlignment(Pos.CENTER);

vbox.getChildren().addAll(headgp1,avaserl);

Scene scene15=new Scene(vbox);

scene15.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog1.setScene(scene15);

dialog1.initStyle(StageStyle.DECORATED);

dialog1.showAndWait();

}

});

btn.setId("btn");

btn.setPrefSize(190, 50);

contentfp.getChildren().addAll(fromcb,tocb,datedp,bustypecb,btn);

contentfp.setAlignment(Pos.CENTER);

StackPane tfsp = new StackPane();

tfsp.getChildren().addAll(imgmainv,contentfp);

VBox headervb = new VBox();

headervb.getChildren().addAll(headgp,tfsp);

bpane.setTop(headervb);

GridPane gpane2 = new GridPane();

Image imgflogo = new Image(getClass().getResourceAsStream("logo.png"));

ImageView imgflogov = new ImageView(imgflogo);

imgflogov.setFitWidth(70);

imgflogov.setFitHeight(70);

Text t1 = new Text("PARADISE TRAVEL");

t1.setFont(new Font("Arial",19));

Text t2 = new Text("Ⓒ 2020 ibibogroup All rights reserved ");

t2.setFont(new Font("Arial",19));

gpane2.setHgap(10);

gpane2.add(imgflogov, 0, 0);

gpane2.add(t1, 1, 0);

gpane2.add(t2, 115, 0);

bpane.setBottom(gpane2);

Scene scene7=new Scene(bpane);

scene7.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog.setScene(scene7);

dialog.initStyle(StageStyle.DECORATED);

dialog.showAndWait();

}

public void canceltick(){

Alert a = new Alert(AlertType.NONE);

final Stage dialog1=new Stage();

dialog1.initModality(Modality.APPLICATION\_MODAL);

VBox vbox1 = new VBox();

GridPane headgp2 = new GridPane();

headgp2.setId("header");

Image imglogo2 = new Image(getClass().getResourceAsStream("logo.png"));

ImageView imglogov2 = new ImageView(imglogo2);

imglogov2.setFitWidth(80);

imglogov2.setFitHeight(100);

Label bustick2 = new Label("CANCEL TICKETS");

bustick2.setId("head");

headgp2.setHgap(10);

headgp2.add(imglogov2, 10, 0);

headgp2.add(bustick2, 14, 0);

GridPane cangp = new GridPane();

cangp.setPadding(new Insets(20));

Label busid = new Label("Bus Id ");

ComboBox busnocb = new ComboBox();

busnocb.getItems().addAll("101","102","103","104","105","106","107","108","109","110","111","112","113","114");

busnocb.setPrefSize(270, 50);

Label seatno = new Label("Seat No ");

TextField seatnotf = new TextField();

cangp.setHgap(10);

cangp.add(busid, 0, 0);

cangp.add(busnocb, 1, 0);

cangp.add(seatno, 3, 0);

cangp.add(seatnotf, 4, 0);

Button canbtn =new Button("Cancel");

canbtn.setOnAction(e->{

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("delete from bookbus where bookbus.busid=? and bookbus.seat=?");

ps.setString(1,busnocb.toString());

ps.setString (2,seatnotf.toString());

ps.executeUpdate();

a.setAlertType(AlertType.CONFIRMATION);

a.setContentText("Are you sure to cancel your ticket");

Optional<ButtonType> result = a.showAndWait();

ButtonType button = result.orElse(ButtonType.CANCEL);

if (button == ButtonType.OK) {

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Ticket Cancelled");

a.showAndWait();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

});

FlowPane cancelbtnfp = new FlowPane(canbtn);

cancelbtnfp.setAlignment(Pos.CENTER);

cancelbtnfp.setPadding(new Insets(20));

vbox1.getChildren().addAll(headgp2,cangp,cancelbtnfp);

Scene = new Scene(vbox1);

scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog1.initStyle(StageStyle.DECORATED);

dialog1.setScene(scene);

dialog1.show();

}

public void booktickets(){

Alert a = new Alert(AlertType.NONE);

final Stage dialog2=new Stage();

dialog2.initModality(Modality.APPLICATION\_MODAL);

VBox vbox1 = new VBox();

GridPane headgp2 = new GridPane();

headgp2.setId("header");

Image imglogo2 = new Image(getClass().getResourceAsStream("logo.png"));

ImageView imglogov2 = new ImageView(imglogo2);

imglogov2.setFitWidth(80);

imglogov2.setFitHeight(100);

Label bustick2 = new Label("BOOK TICKETS");

bustick2.setId("head");

headgp2.setHgap(10);

headgp2.add(imglogov2, 20, 0);

headgp2.add(bustick2, 22, 0);

HBox = new HBox();

GridPane bookgp = new GridPane();

Label busid = new Label("Bus Id ");

TextField busidtf = new TextField();

Label custname = new Label("Customer Name ");

TextField custnametf = new TextField();

Label seatno = new Label("Seat No ");

TextField seatnotf = new TextField();

Label mobno = new Label("Mobile No ");

TextField mobnotf = new TextField();

Label datel = new Label("Date ");

TextField datetf = new TextField();

Button addbtn = new Button("Add");

addbtn.setOnAction(e2 -> {

String idbus=busidtf.getText();

String seatnumber=seatnotf.getText();

String customername=custnametf.getText();

String mobileno=mobnotf.getText();

String date=datetf.getText();

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("insert into BOOKBUS(busid,seat,customer,mobileno,date)values(?,?,?,?,?)");

ps.setString(1,idbus);

ps.setString (2,seatnumber);

ps.setString(3,customername);

ps.setString(4,mobileno);

ps.setString(5,date);

ps.executeUpdate();

PreparedStatement ps1=con.prepareStatement("update seat set status=? where seats=?");

ps1.setString(1, "Booked");

ps1.setString(2, seatnumber);

ps1.executeUpdate();

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Bus Booked");

Optional<ButtonType> result = a.showAndWait();

ButtonType button = result.orElse(ButtonType.CANCEL);

if (button == ButtonType.OK) {

refresh();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

});

Button canbtn = new Button("Cancel");

bookgp.setHgap(10);

bookgp.setVgap(10);

bookgp.add(busid,0,0);

bookgp.add(busidtf,1,0);

bookgp.add(custname,0,1);

bookgp.add(custnametf,1,1);

bookgp.add(seatno,0,2);

bookgp.add(seatnotf,1,2);

bookgp.add(mobno,0,3);

bookgp.add(mobnotf,1,3);

bookgp.add(datel,0,4);

bookgp.add(datetf,1,4);

bookgp.add(addbtn,0,5);

bookgp.add(canbtn,1,5);

bookgp.setPadding(new Insets(20));

tableview = new TableView();

//tableview.setMaxSize(800, 800);

tableview.getSelectionModel().setSelectionMode(SelectionMode.SINGLE);

booking();

tableview.getSelectionModel().selectedItemProperty().addListener(new ChangeListener() {

public void changed(ObservableValue, Object oldValue, Object newValue) {

if(tableview.getSelectionModel().getSelectedItem() != null)

{

TableViewSelectionModel selectionModel = tableview.getSelectionModel();

ObservableList selectedCells = selectionModel.getSelectedCells();

TablePosition = (TablePosition) selectedCells.get(0);

Object val = tablePosition.getTableColumn().getCellData(tablePosition.getRow());

seatnotf.setText(val.toString());

}

}

});

tableview.setPadding(new Insets(20));

hbox.getChildren().addAll(bookgp,tableview);

hbox.setPadding(new Insets(20));

vbox1.getChildren().addAll(headgp2,hbox);

//Main Scene

Scene = new Scene(vbox1);

scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog2.initStyle(StageStyle.DECORATED);

dialog2.setScene(scene);

dialog2.show();

}

public void refresh(){

Alert a = new Alert(AlertType.NONE);

final Stage dialog2=new Stage();

dialog2.initModality(Modality.APPLICATION\_MODAL);

VBox vbox1 = new VBox();

GridPane headgp2 = new GridPane();

headgp2.setId("header");

Image imglogo2 = new Image(getClass().getResourceAsStream("logo.png"));

ImageView imglogov2 = new ImageView(imglogo2);

imglogov2.setFitWidth(80);

imglogov2.setFitHeight(100);

Label bustick2 = new Label("BOOK TICKETS");

bustick2.setId("head");

headgp2.setHgap(10);

headgp2.add(imglogov2, 20, 0);

headgp2.add(bustick2, 22, 0);

HBox = new HBox();

GridPane bookgp = new GridPane();

Label busid = new Label("Bus Id ");

TextField busidtf = new TextField();

Label custname = new Label("Customer Name ");

TextField custnametf = new TextField();

Label seatno = new Label("Seat No ");

TextField seatnotf = new TextField();

Label mobno = new Label("Mobile No ");

TextField mobnotf = new TextField();

Label datel = new Label("Date ");

TextField datetf = new TextField();

Button addbtn = new Button("Add");

addbtn.setOnAction(e2 -> {

String idbus=busidtf.getText();

String seatnumber=seatnotf.getText();

String customername=custnametf.getText();

String mobileno=mobnotf.getText();

String date=datetf.getText();

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

PreparedStatement ps=con.prepareStatement("insert into BOOKBUS(busid,seat,customer,mobileno,date)values(?,?,?,?,?)");

ps.setString(1,idbus);

ps.setString (2,seatnumber);

ps.setString(3,customername);

ps.setString(4,mobileno);

ps.setString(5,date);

ps.executeUpdate();

PreparedStatement ps1=con.prepareStatement("update seat set seats=? where seats=?");

ps1.setString(1, "Booked");

ps1.setString(2, seatnumber);

ps1.executeUpdate();

a.setAlertType(AlertType.INFORMATION);

a.setContentText("Login Successful");

Optional<ButtonType> result = a.showAndWait();

ButtonType button = result.orElse(ButtonType.CANCEL);

if (button == ButtonType.OK) {

//refresh();

}

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

});

Button canbtn = new Button("Cancel");

bookgp.setHgap(10);

bookgp.setVgap(10);

bookgp.add(busid,0,0);

bookgp.add(busidtf,1,0);

bookgp.add(custname,0,1);

bookgp.add(custnametf,1,1);

bookgp.add(seatno,0,2);

bookgp.add(seatnotf,1,2);

bookgp.add(mobno,0,3);

bookgp.add(mobnotf,1,3);

bookgp.add(datel,0,4);

bookgp.add(datetf,1,4);

bookgp.add(addbtn,0,5);

bookgp.add(canbtn,1,5);

bookgp.setPadding(new Insets(20));

tableview = new TableView();

tableview.setMaxSize(800, 800);

tableview.getSelectionModel().setSelectionMode(SelectionMode.SINGLE);

booking();

tableview.getSelectionModel().selectedItemProperty().addListener(new ChangeListener() {

public void changed(ObservableValue, Object oldValue, Object newValue) {

if(tableview.getSelectionModel().getSelectedItem() != null)

{

TableViewSelectionModel selectionModel = tableview.getSelectionModel();

ObservableList selectedCells = selectionModel.getSelectedCells();

TablePosition = (TablePosition) selectedCells.get(0);

Object val = tablePosition.getTableColumn().getCellData(tablePosition.getRow());

seatnotf.setText(val.toString());

}

}

});

tableview.setPadding(new Insets(20));

hbox.getChildren().addAll(bookgp,tableview);

hbox.setPadding(new Insets(20));

vbox1.getChildren().addAll(headgp2,hbox);

//Main Scene

Scene = new Scene(vbox1);

scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

dialog2.initStyle(StageStyle.DECORATED);

dialog2.setScene(scene);

dialog2.show();

}

public void booking(){

data = FXCollections.observableArrayList();

try{

Class.forName("org.apache.derby.jdbc.ClientDriver");

Connection con=DriverManager.getConnection("jdbc:derby://localhost:1527/Bus\_Ticket\_Reservation","andriya","19ucsa209");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from seat");

for(int i=0 ; i<res.getMetaData().getColumnCount(); i++){

final int j = i;

TableColumn col = new TableColumn(res.getMetaData().getColumnName(i+1));

col.setCellValueFactory(new Callback<CellDataFeatures<ObservableList,String>,ObservableValue<String>>(){

public ObservableValue<String> call(CellDataFeatures<ObservableList, String> param) {

return new SimpleStringProperty(param.getValue().get(j).toString());

}

});

tableview.getColumns().addAll(col);

System.out.println("Column ["+i+"] ");

}

while(res.next()){

ObservableList<String> row = FXCollections.observableArrayList();

for(int i=1 ; i<=res.getMetaData().getColumnCount(); i++){

row.add(res.getString(i));

}

System.out.println("Row [1] added "+row );

data.add(row);

}

tableview.setItems(data);

}

catch(ClassNotFoundException | SQLException err){

System.out.println(err);

}

}

}

style.css

body{

-fx-font-family: Arial;

}

Label{

-fx-font-weight: bold;

-fx-font-size: 20px;

-fx-text-fill :black;

}

#label1{

-fx-font-weight: normal;

-fx-font-size: 20px;

-fx-text-fill :white;

}

.text-field:focused , .combo-box:focused{

-fx-faint-focus-color: transparent;

-fx-focus-color: #df4f56;

}

.button{

-fx-background-color: transparent;

-fx-border-width: 2px;

-fx-border-color:#df4f56;

-fx-text-fill: black;

-fx-font-weight: bold;

}

#loginbtn:pressed , #newloginbtn:pressed , #adminloginbtn:pressed , #asaddbtn:pressed, #ascbtn:pressed , #btn{

-fx-background-color: #df4f56;

-fx-text-fill: white;

}

#header{

-fx-background-color: #df4f56;

-fx-text-fill:white;

}

#footer{

-fx-background-color: #717173;

-fx-text-fill:white;

}

.combo-box .list-cell

{

-fx-background: white;

-fx-background-color: transparent;

-fx-text-fill: black;

}

.combo-box-base:focused {

-fx-background-color: white;

-fx-border-color:#df4f56;

}

.menu-bar{

-fx-font-weight: bold;

-fx-background-color: transparent;

-fx-font-size: 20px;

-fx-text-fill: white;

}

.menu-bar .label {

-fx-text-fill: white;

-fx-background-color: transparent;

}

.context-menu {

-fx-background-color: white;

-fx-font-size: 18px;

-fx-font-weight: bold;

}

.menu-item .label {

-fx-background-color: transparent;

-fx-text-fill: black;

-fx-font-weight: normal;

}

#fromlocicon , #tolocicon {

-fx-background-image:url('loc.png');

-fx-background-repeat: no-repeat;

-fx-background-position: left center;

-fx-background-size: 25px;

-fx-font-size: 19;

}

#head{

-fx-font-weight: bold;

-fx-font-size: 20px;

-fx-text-fill :white;

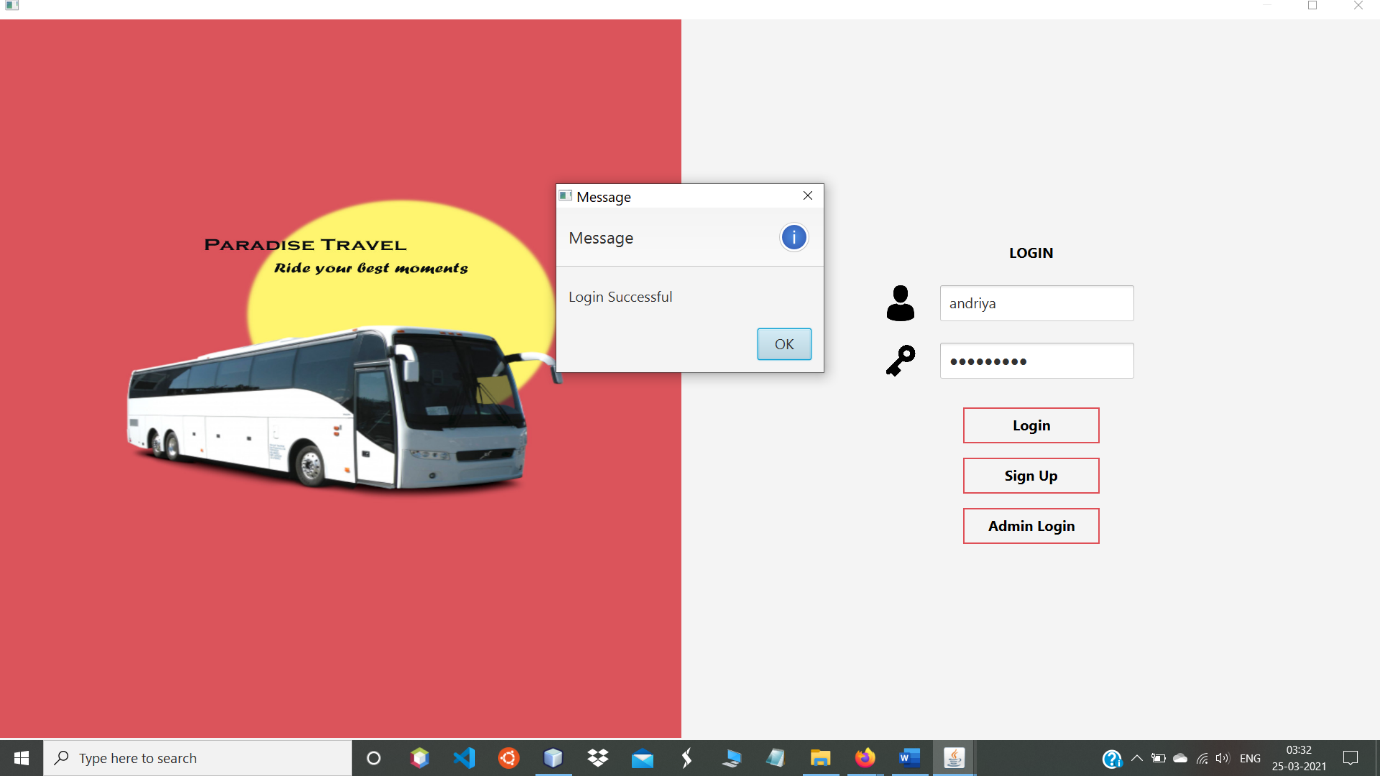
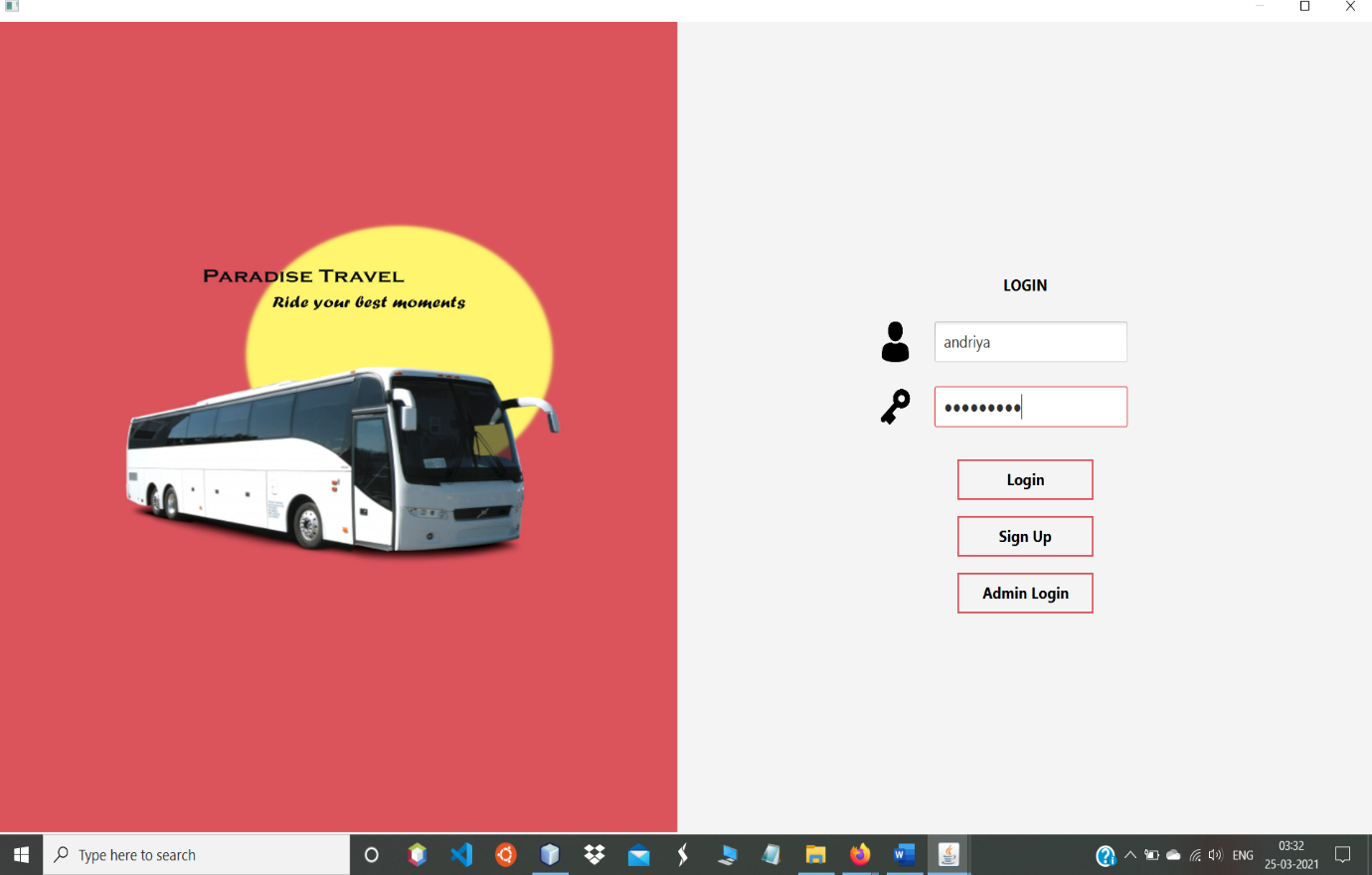
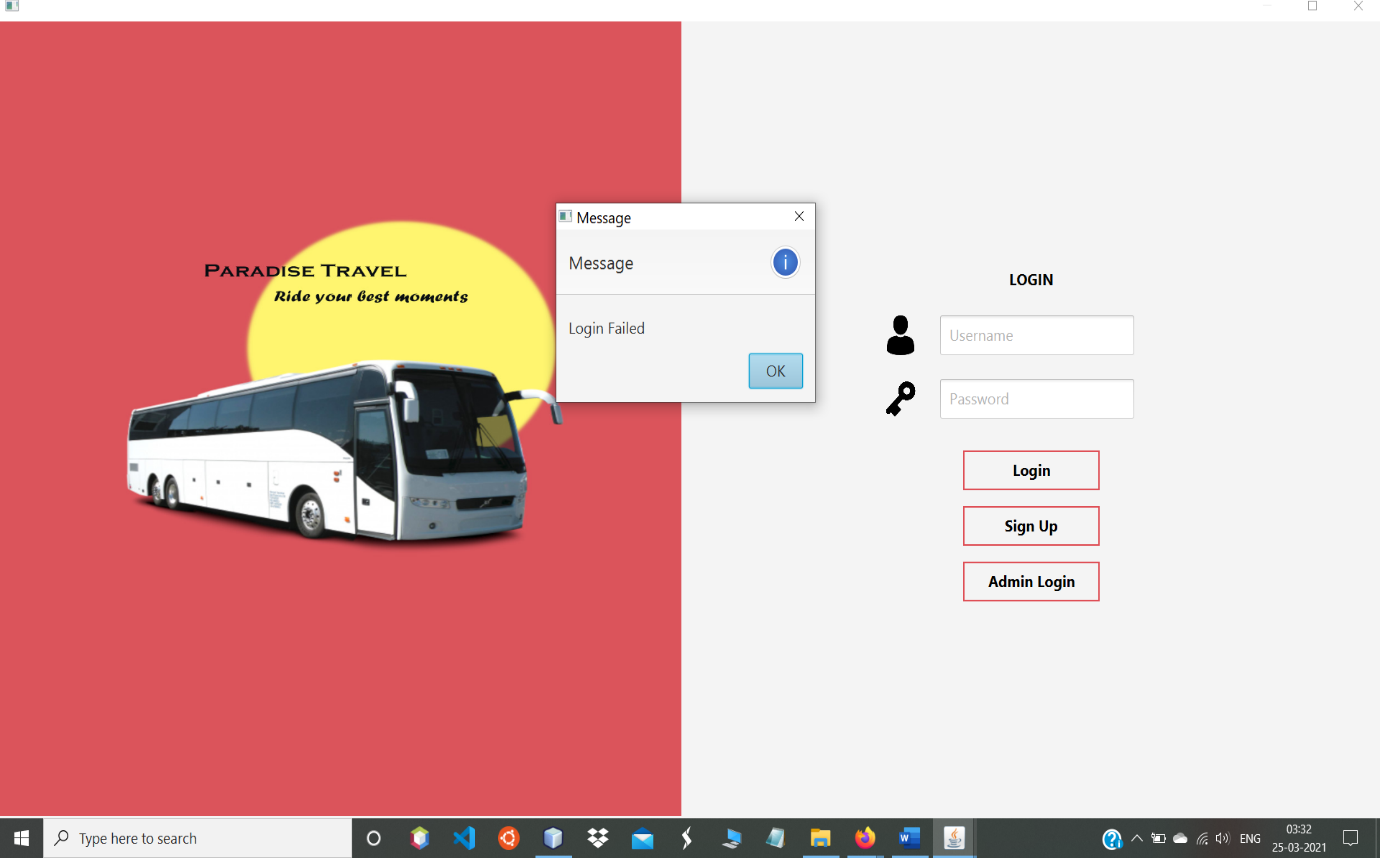
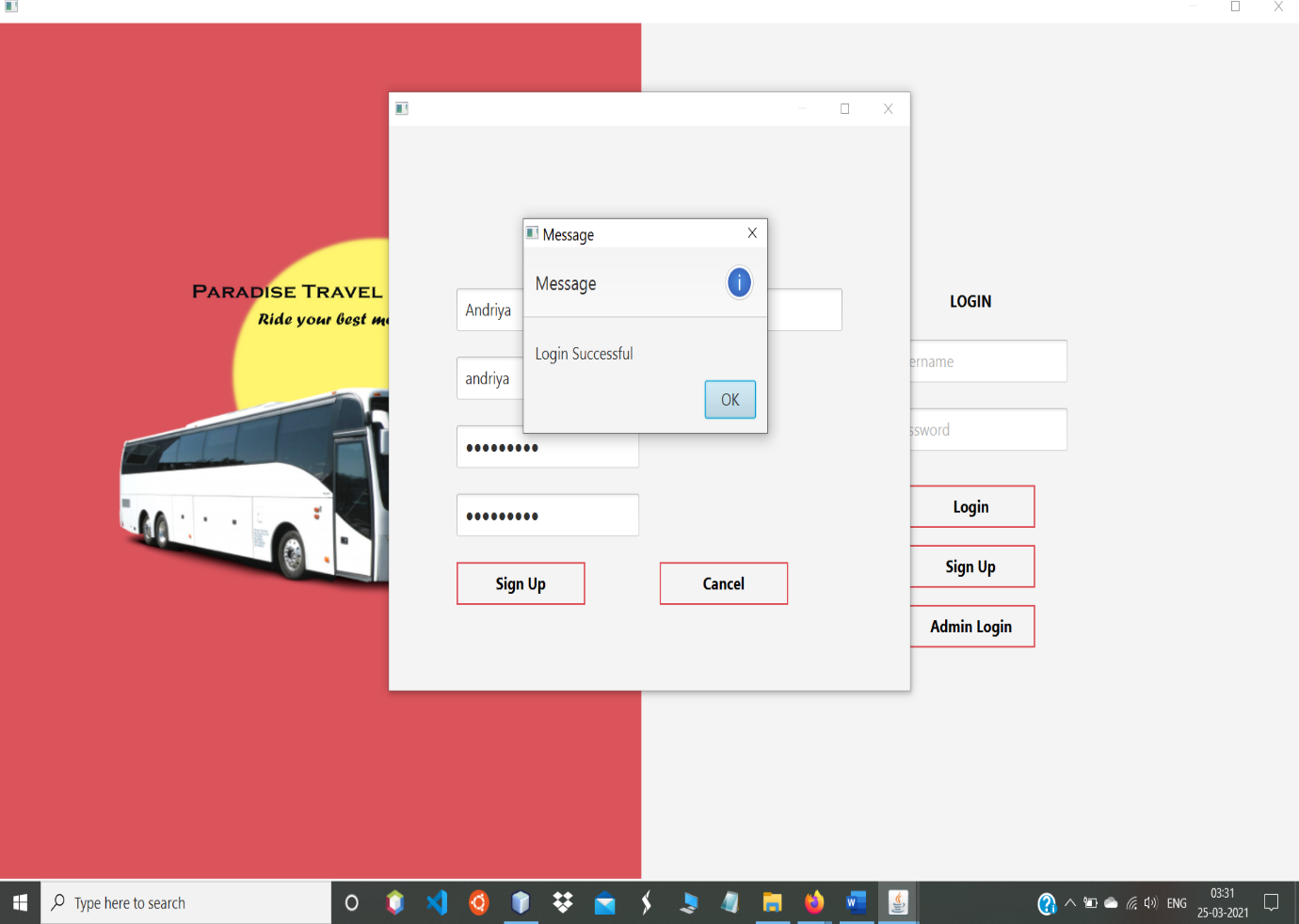
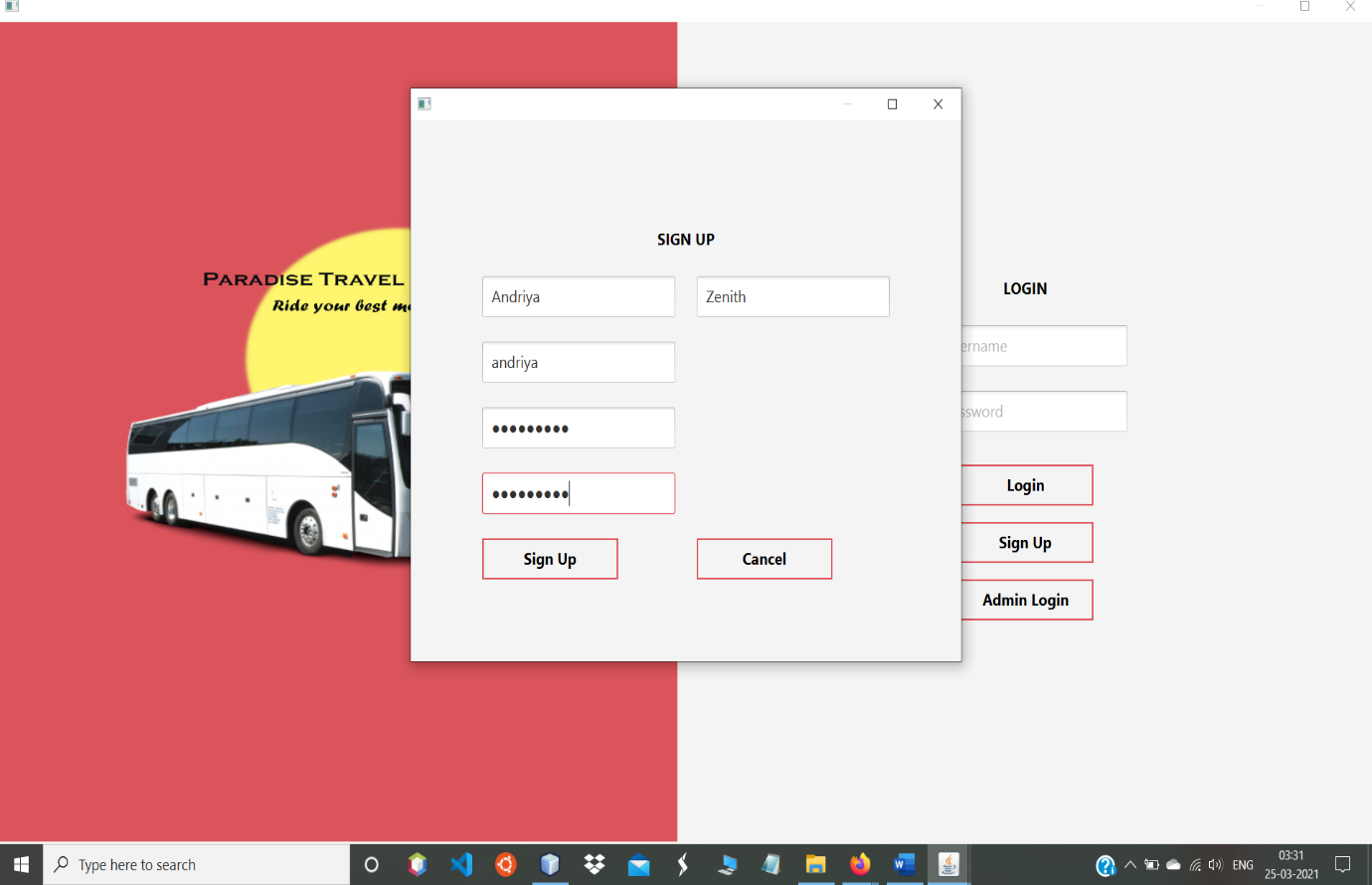
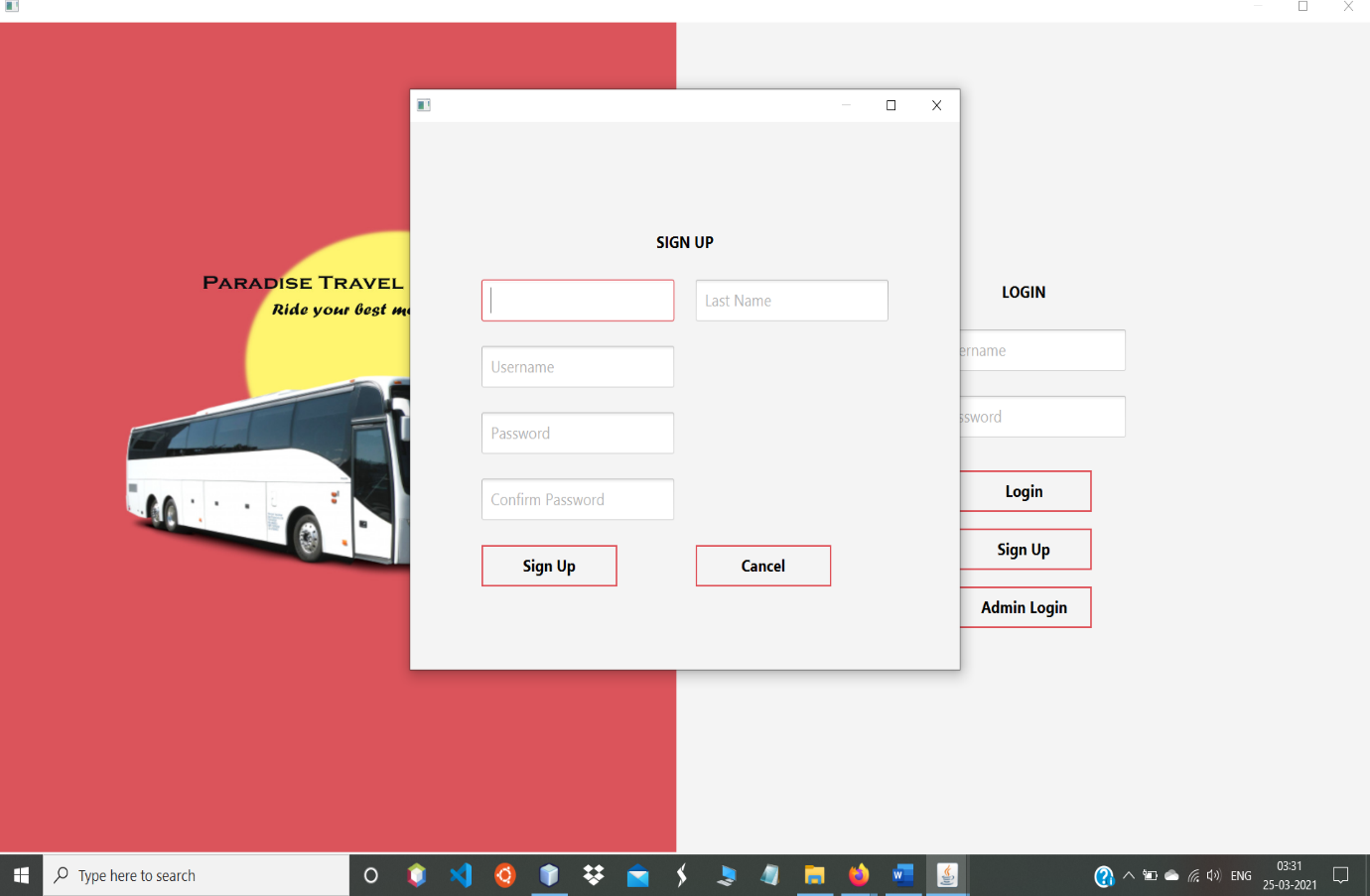
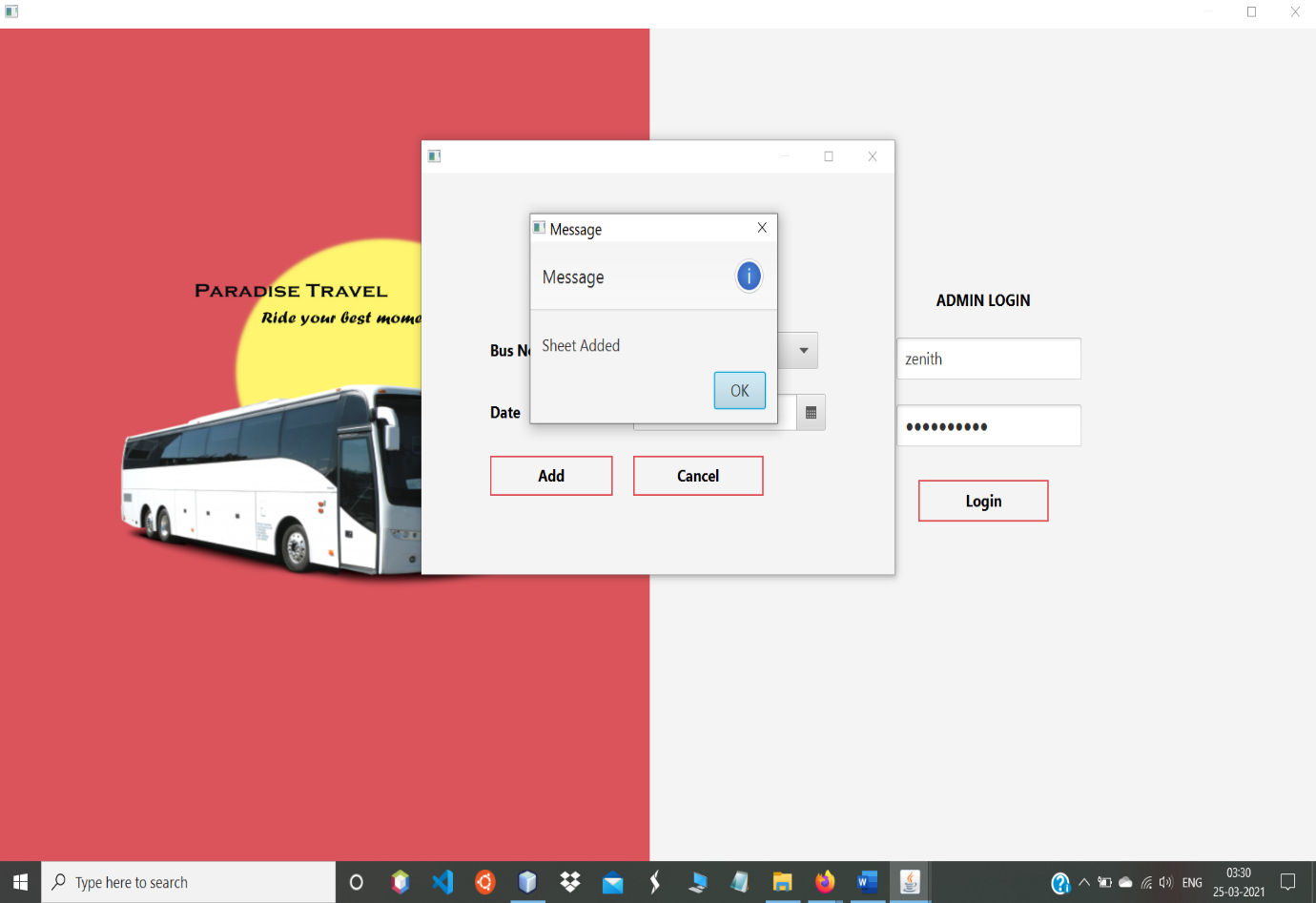
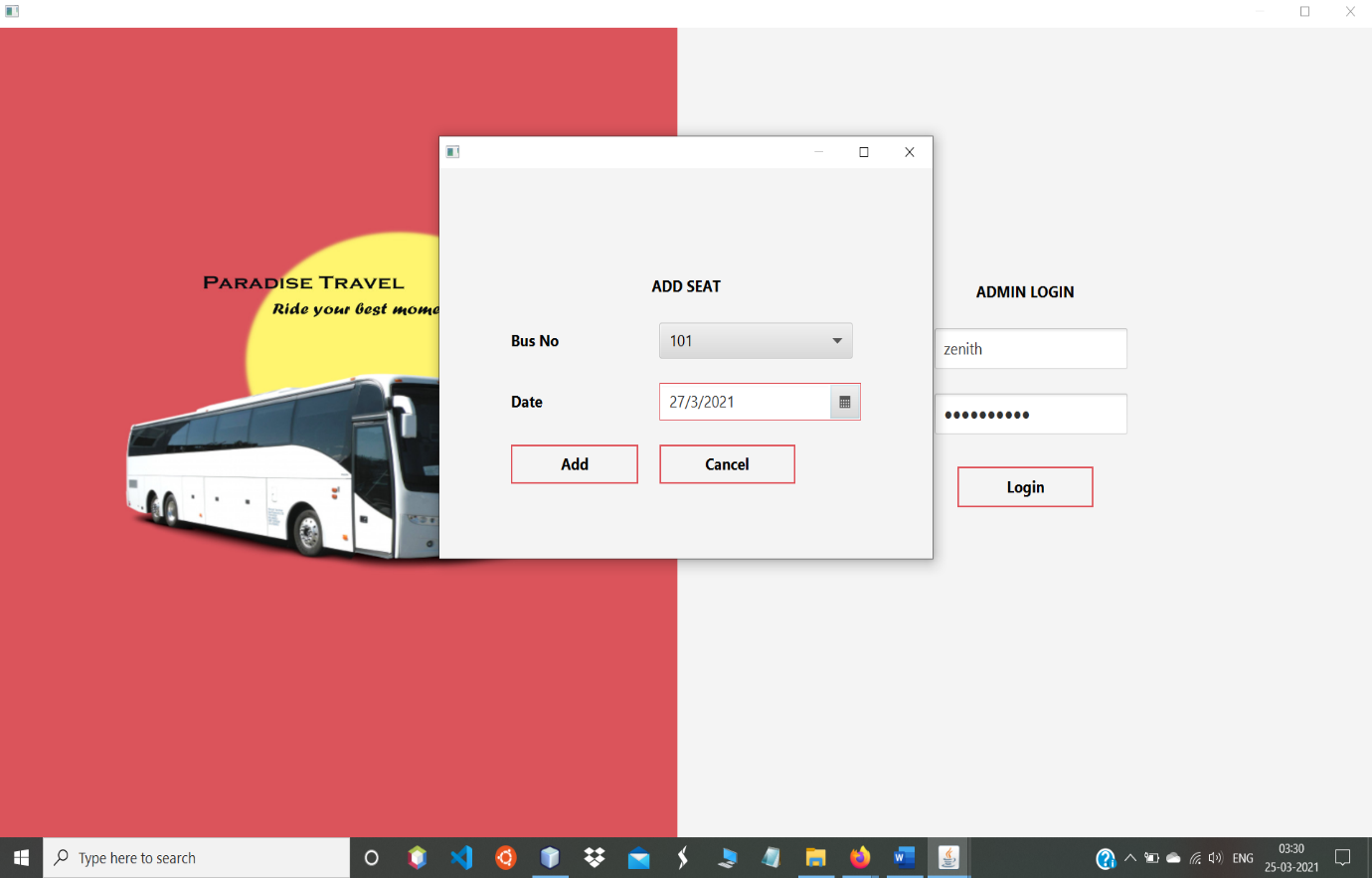
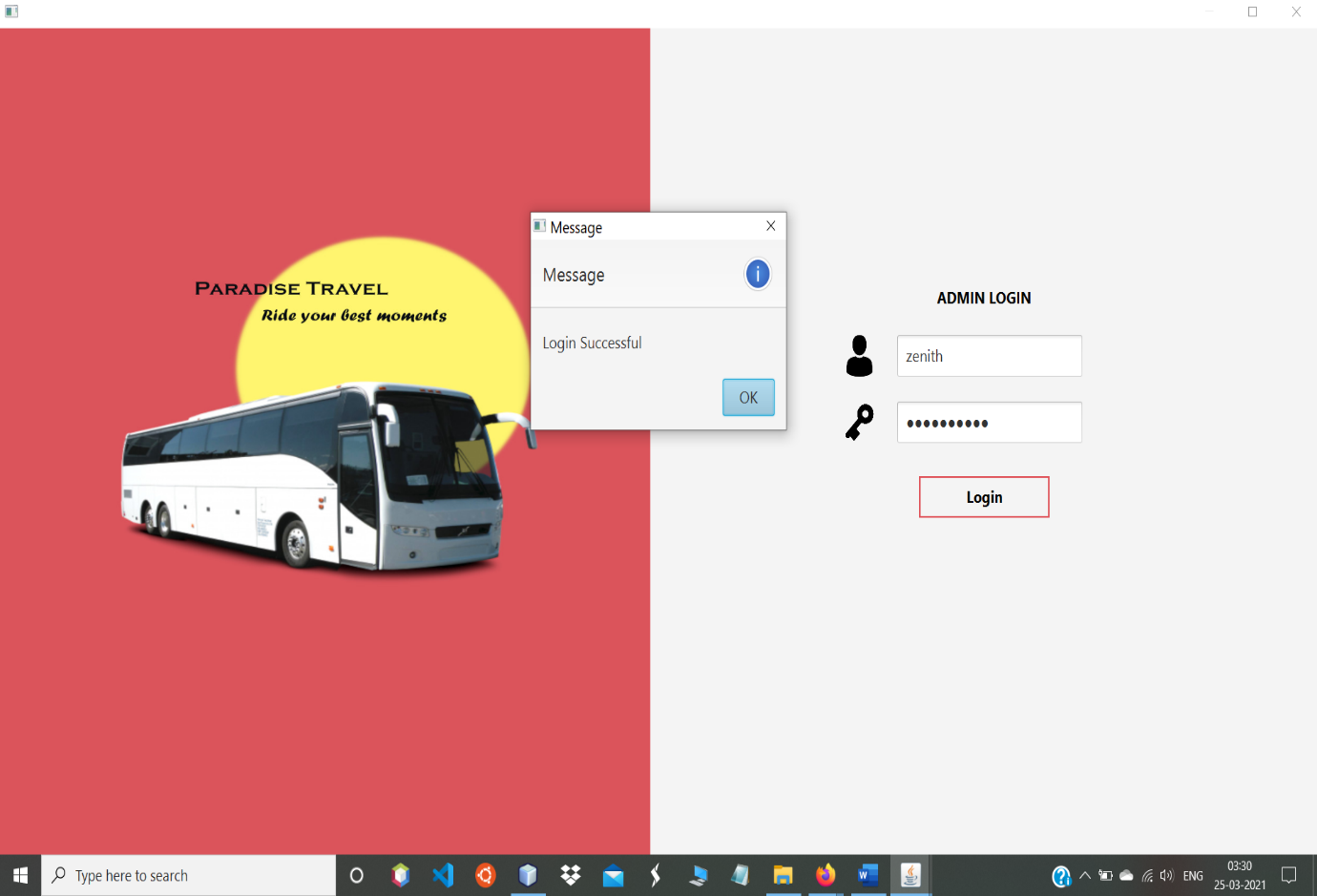
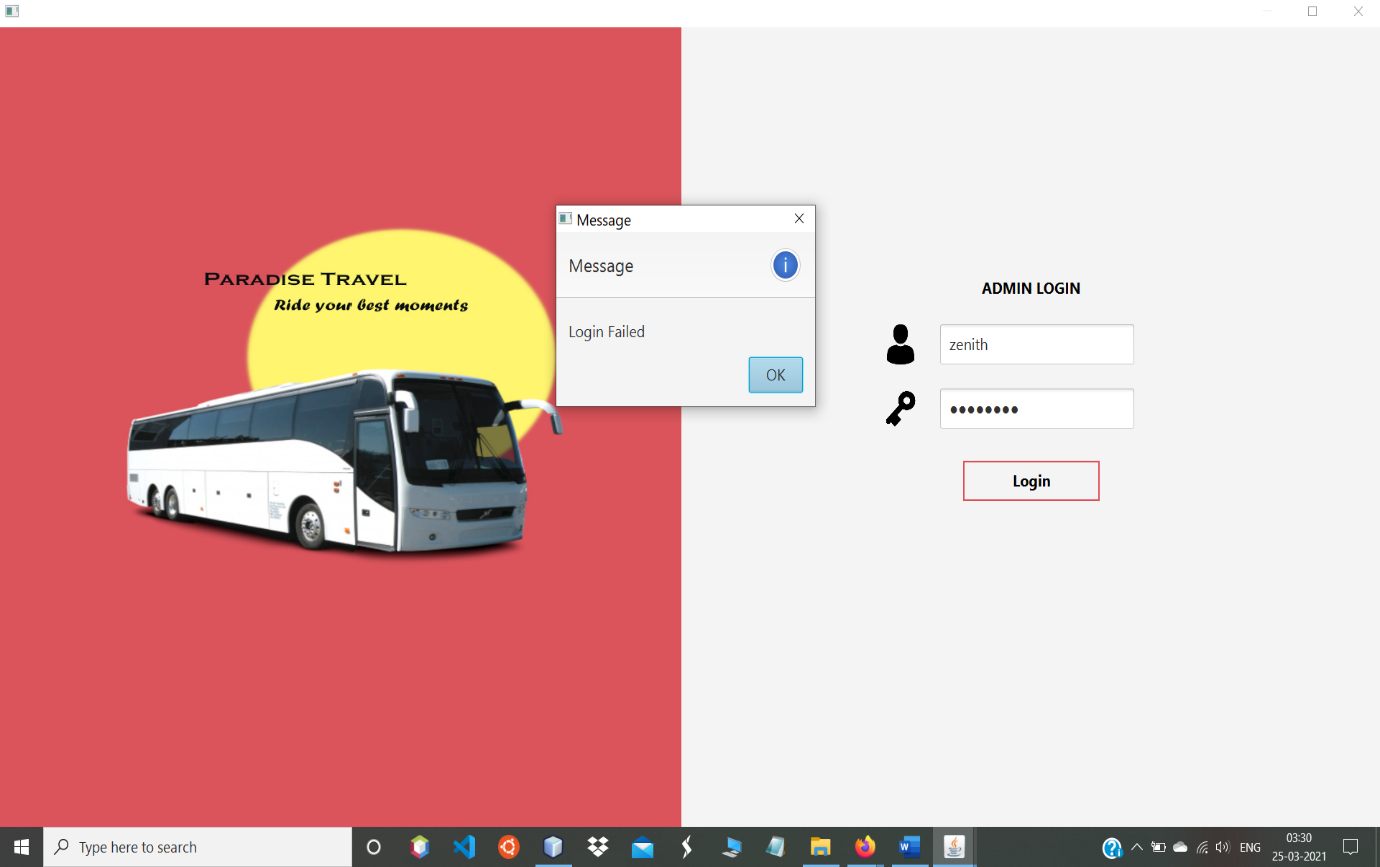
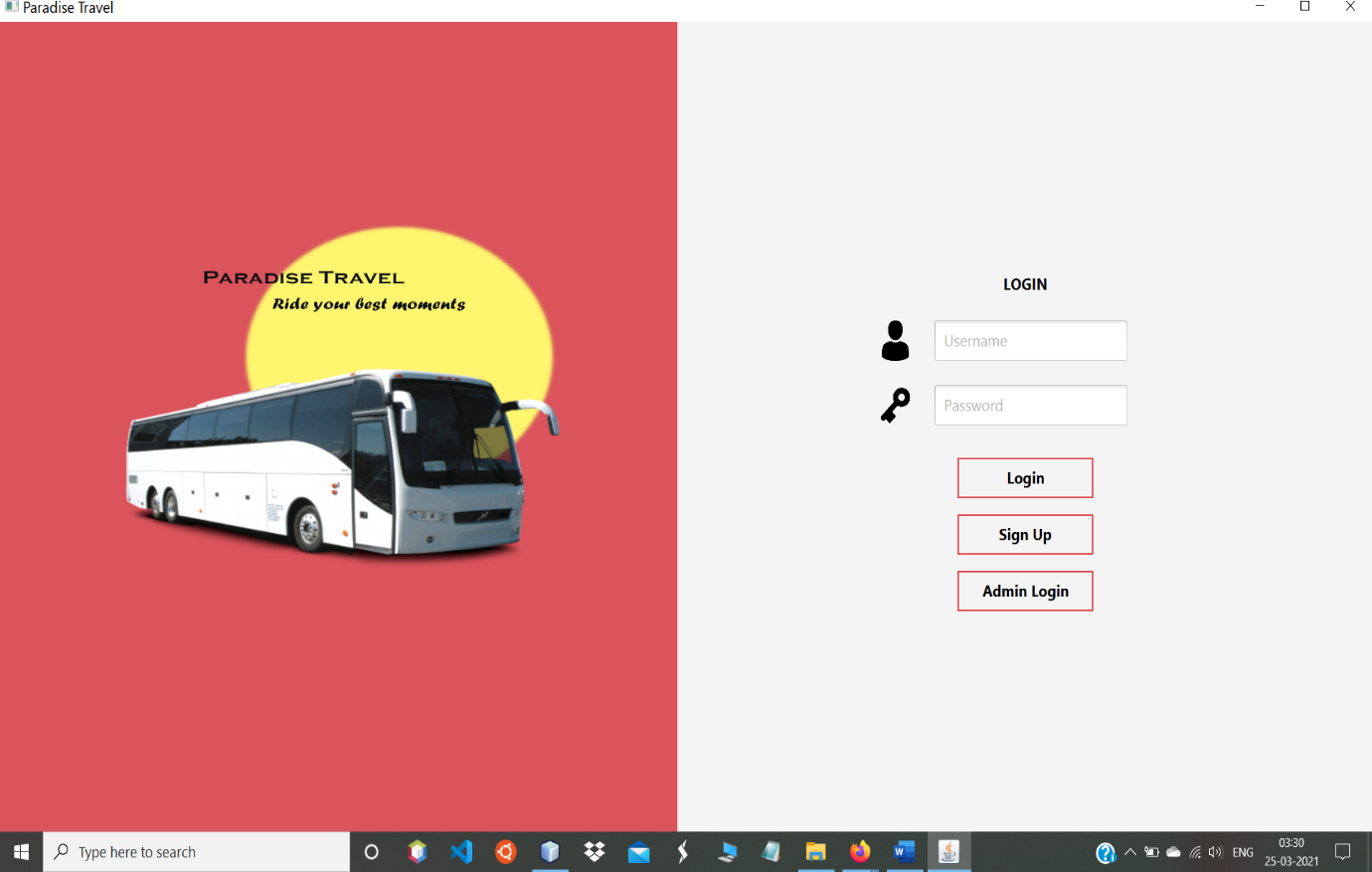
}

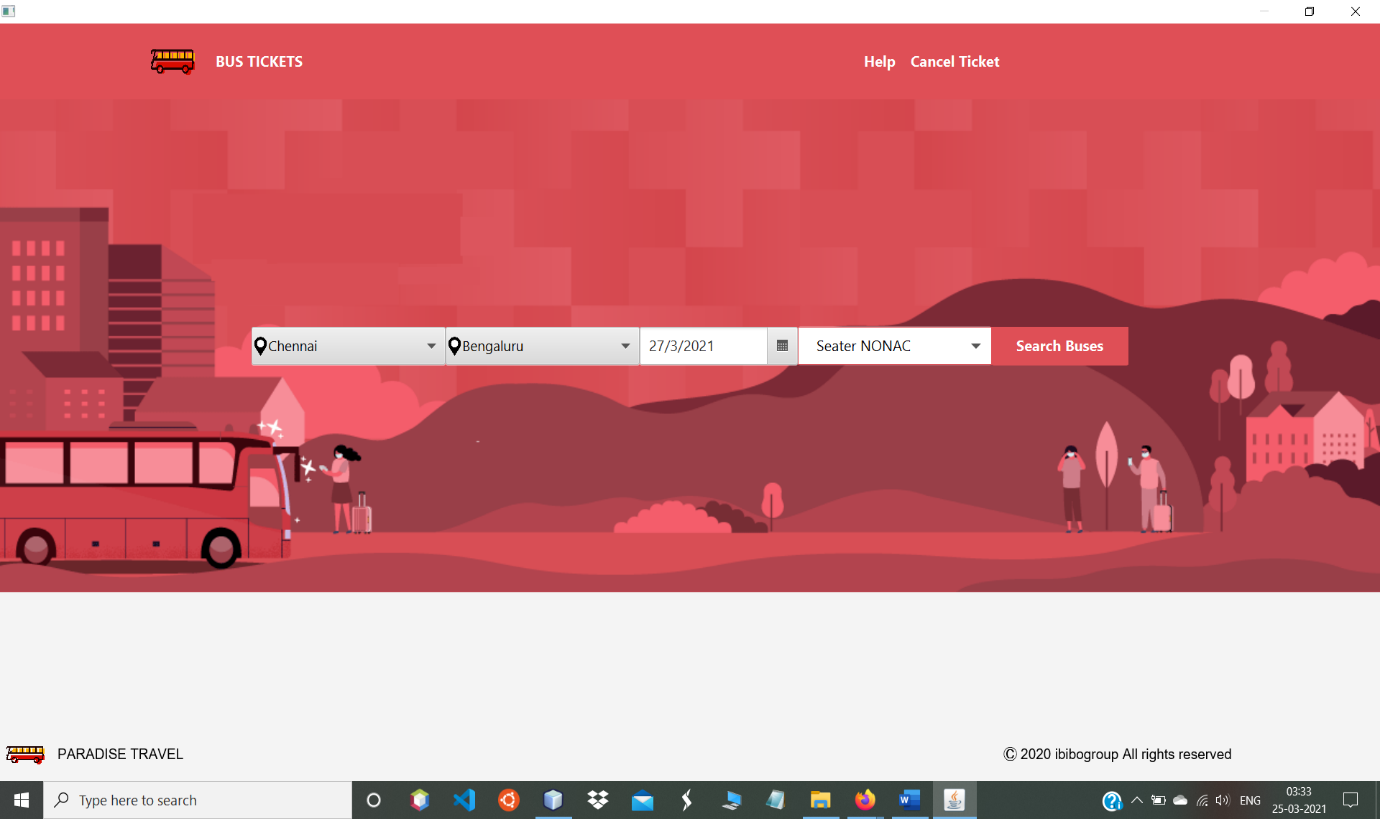
.table-row-cell {

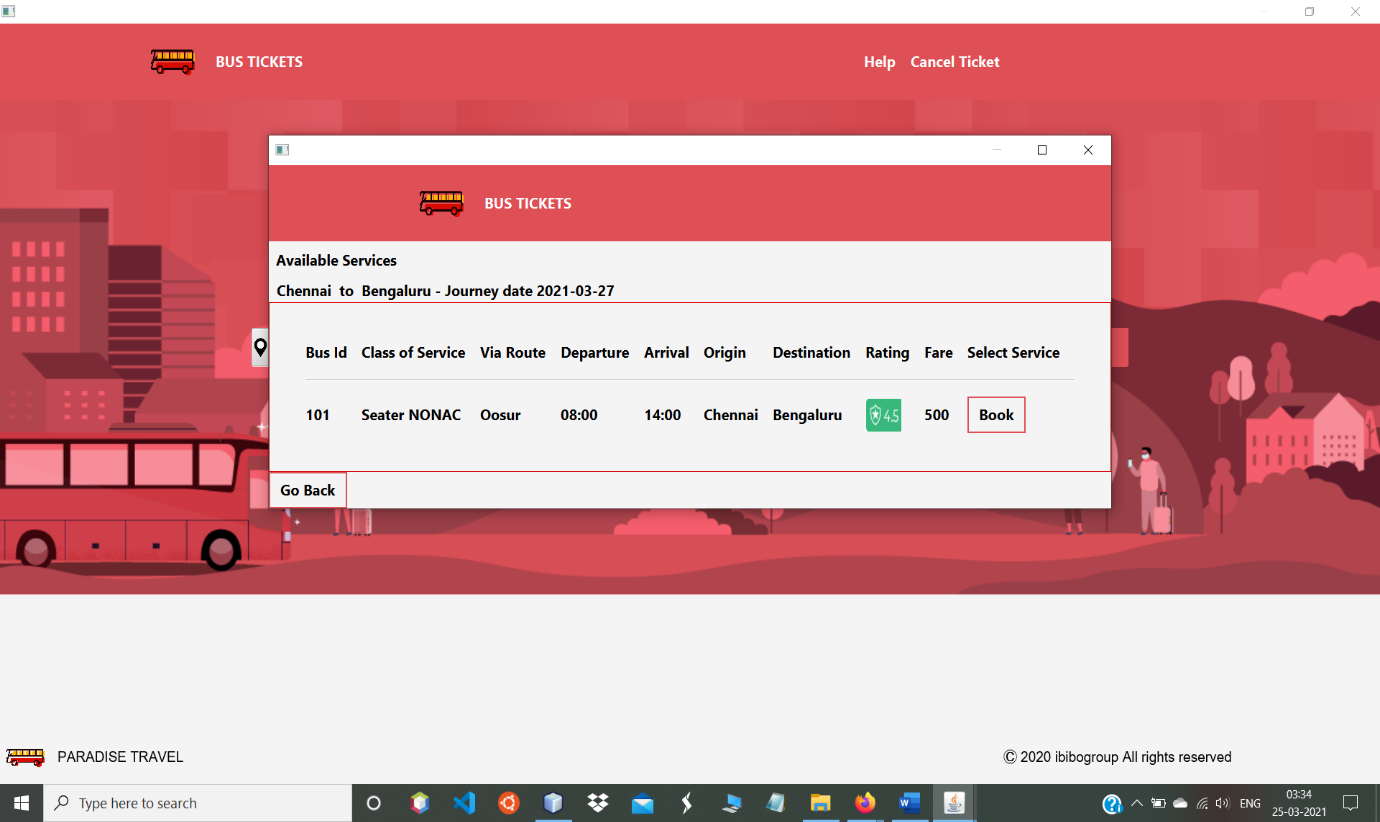
-fx-cell-size: 70px;

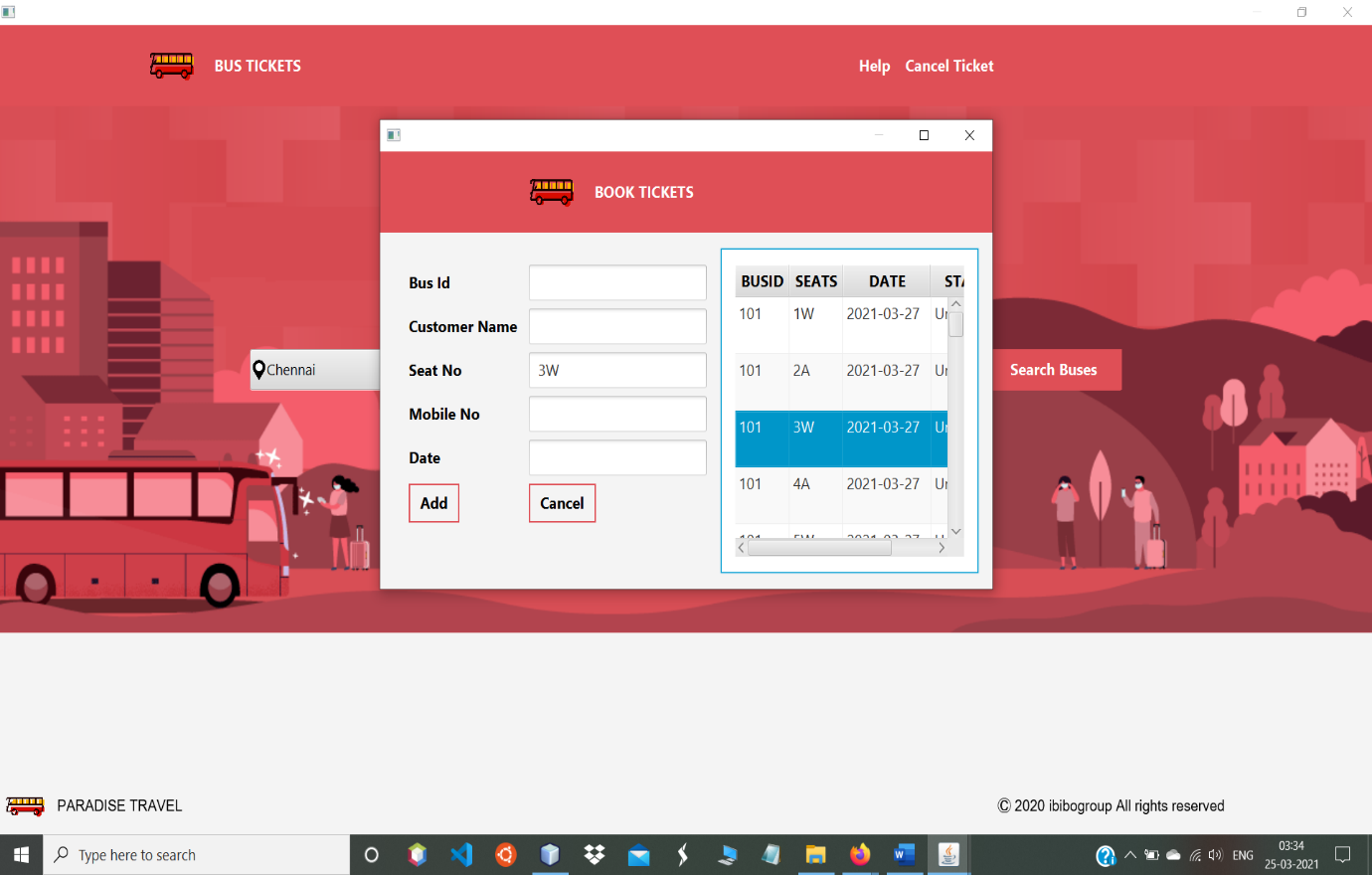
}

**OUTPUT**

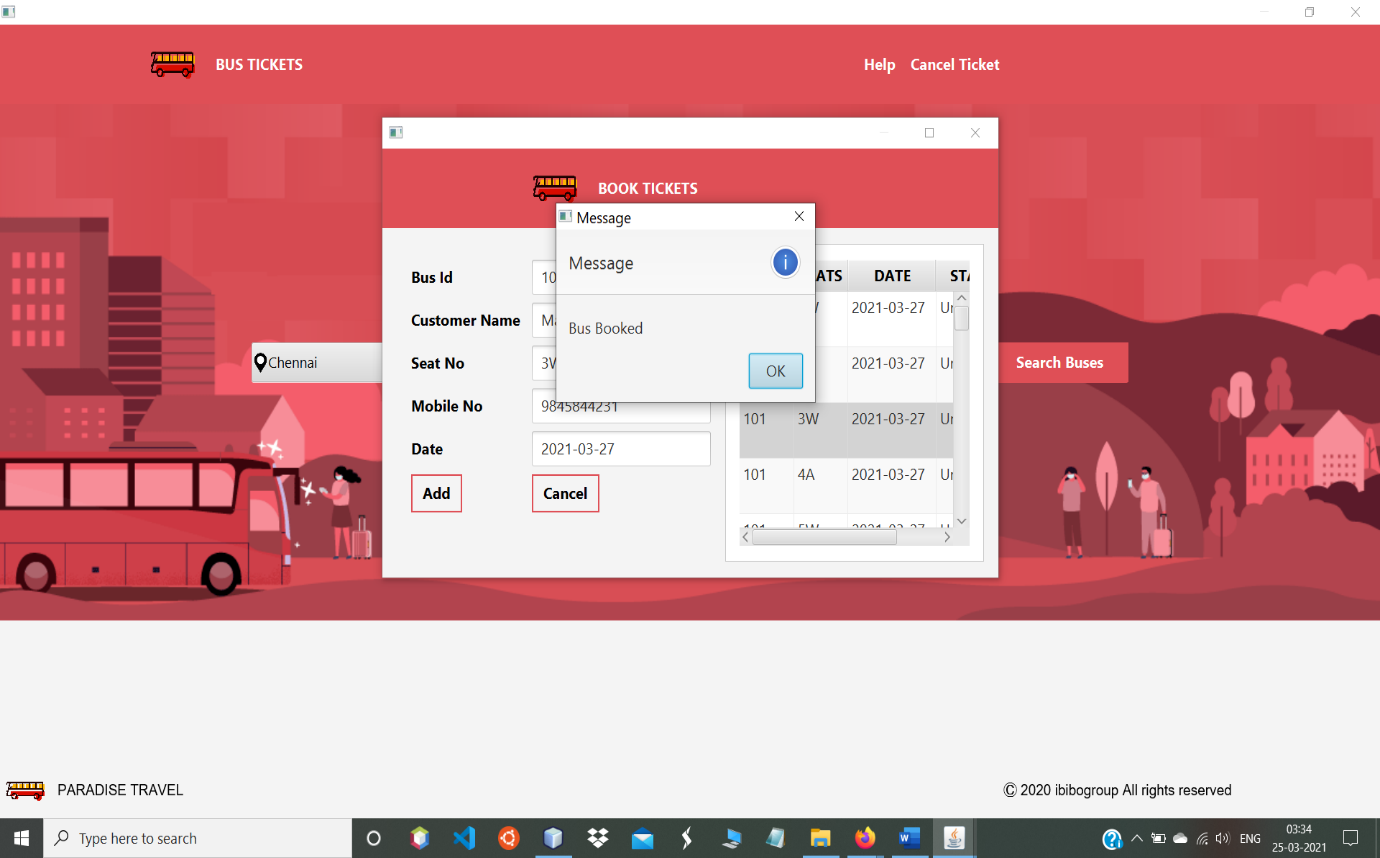
****

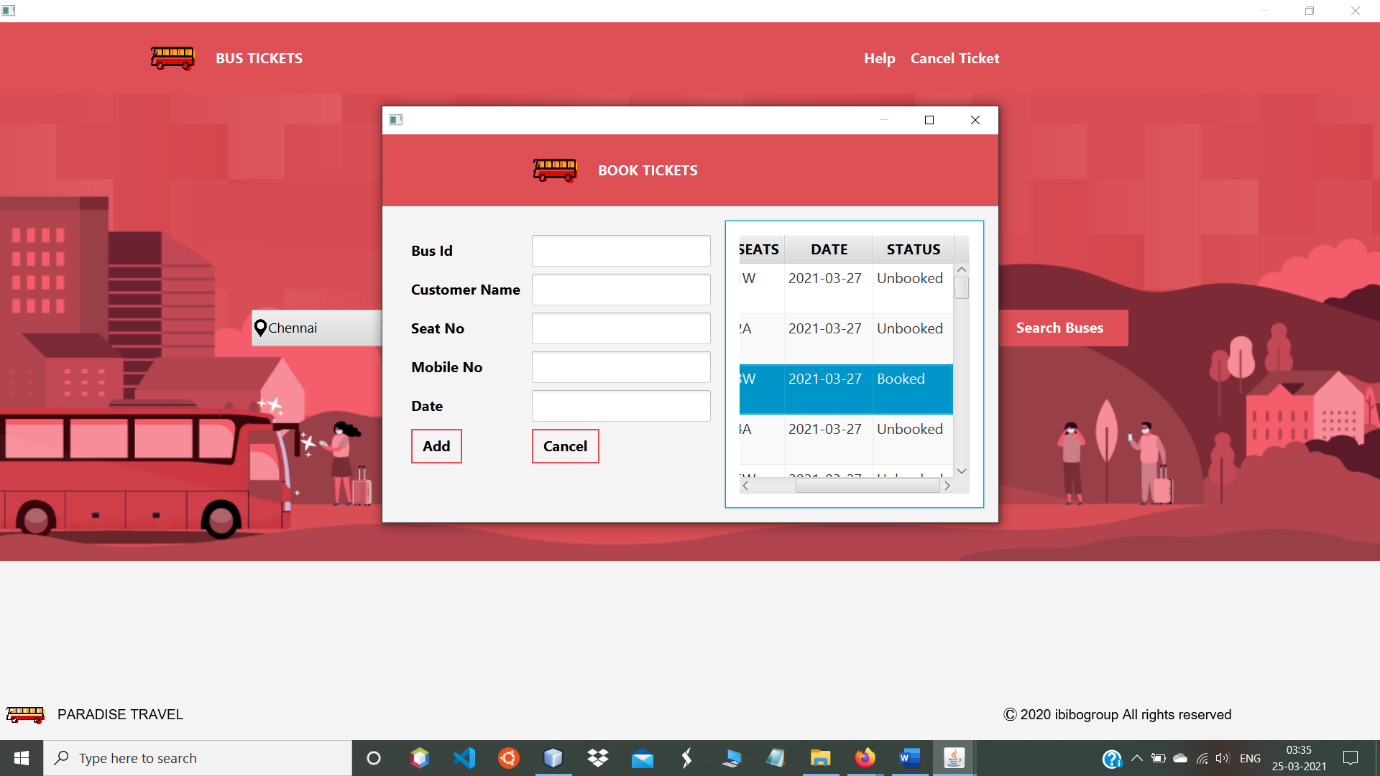
****

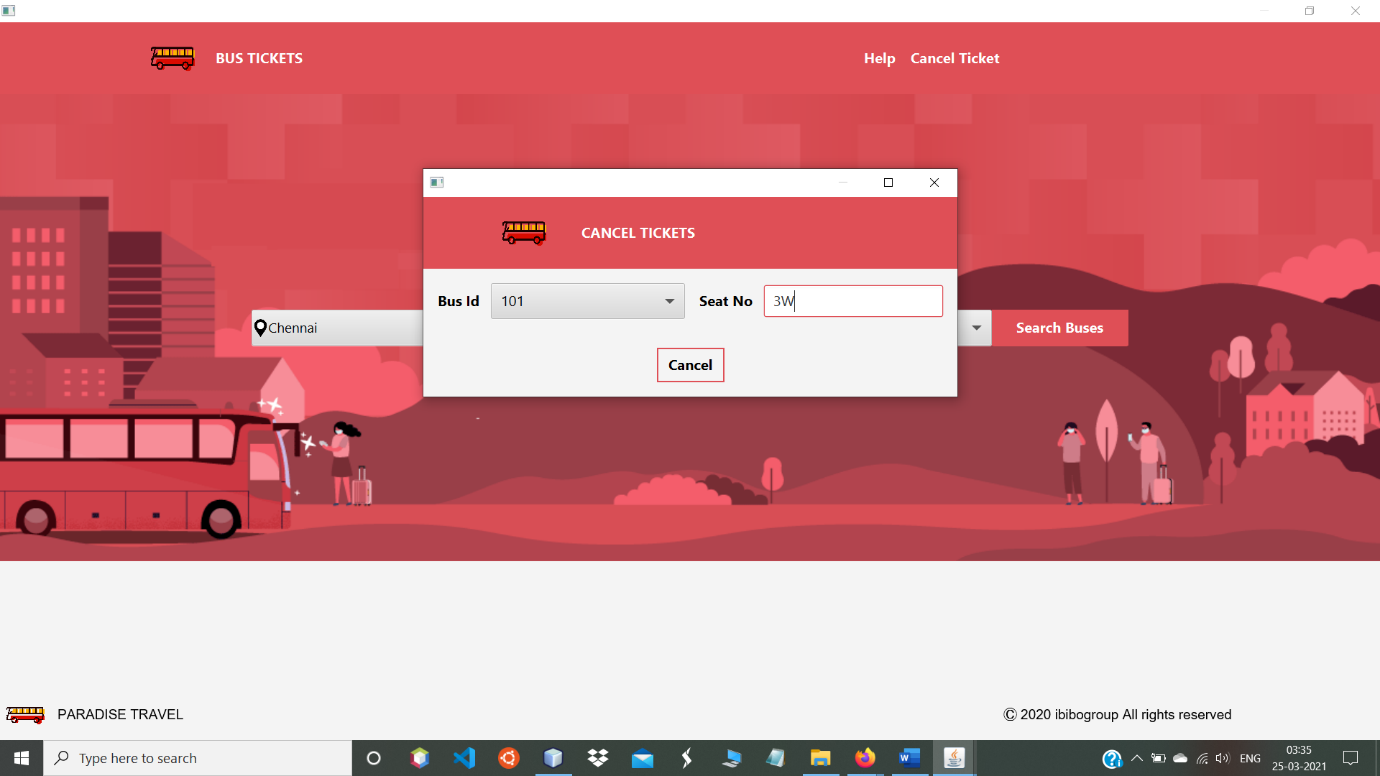
****

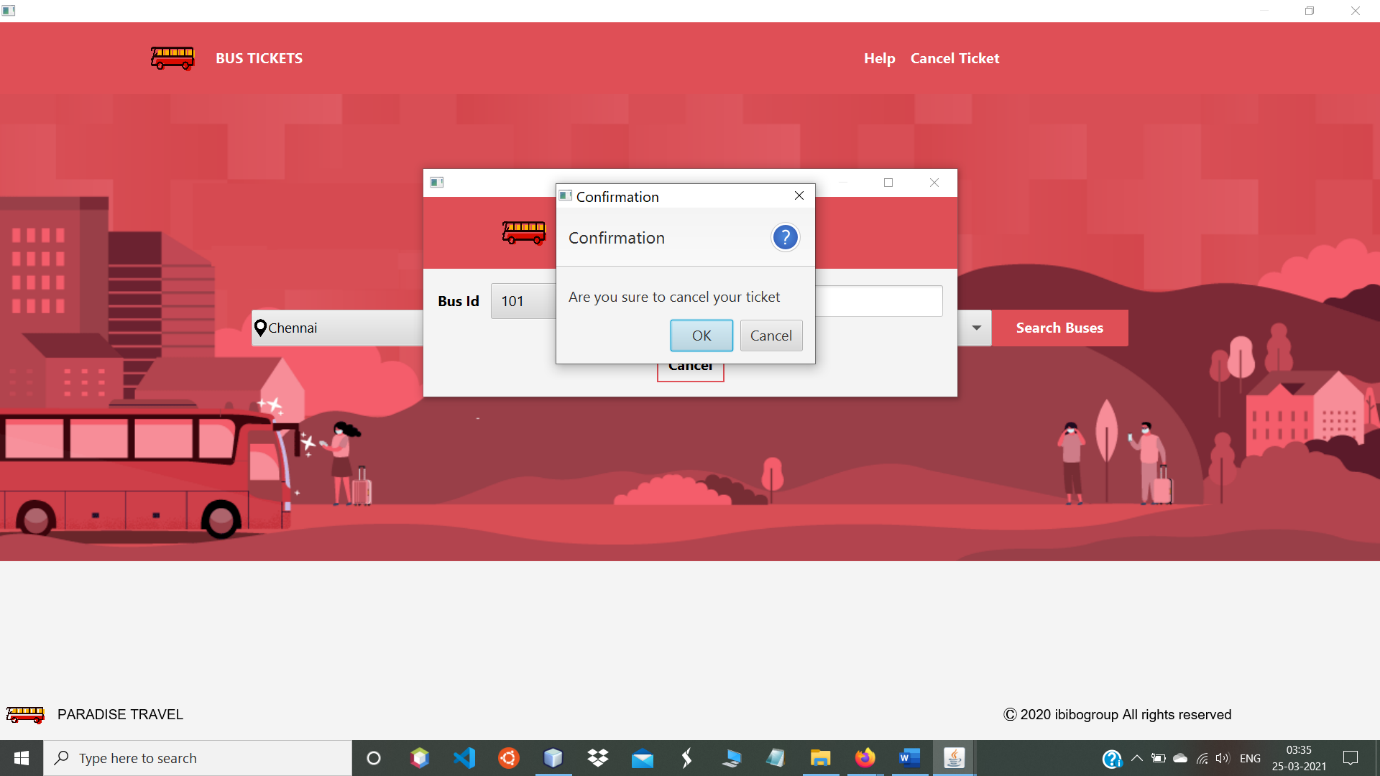
****

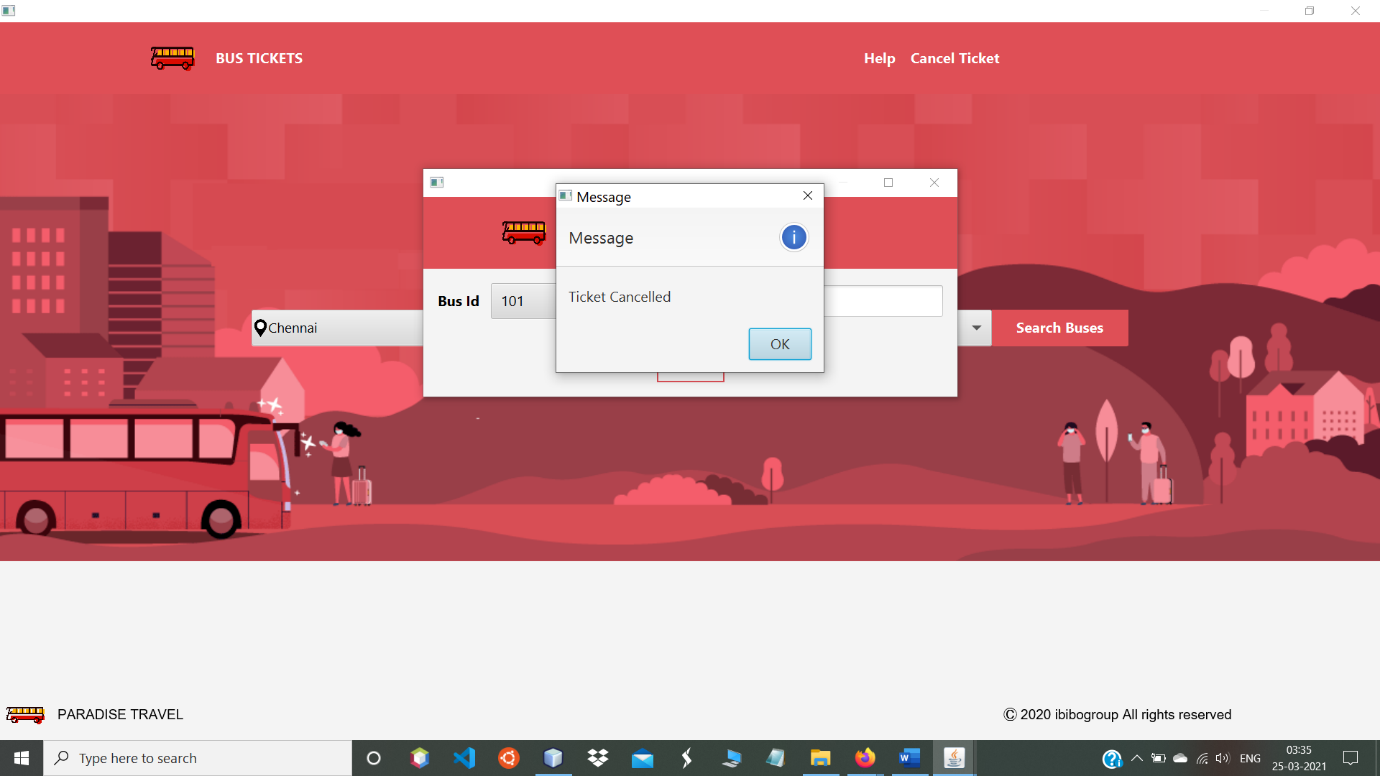
****

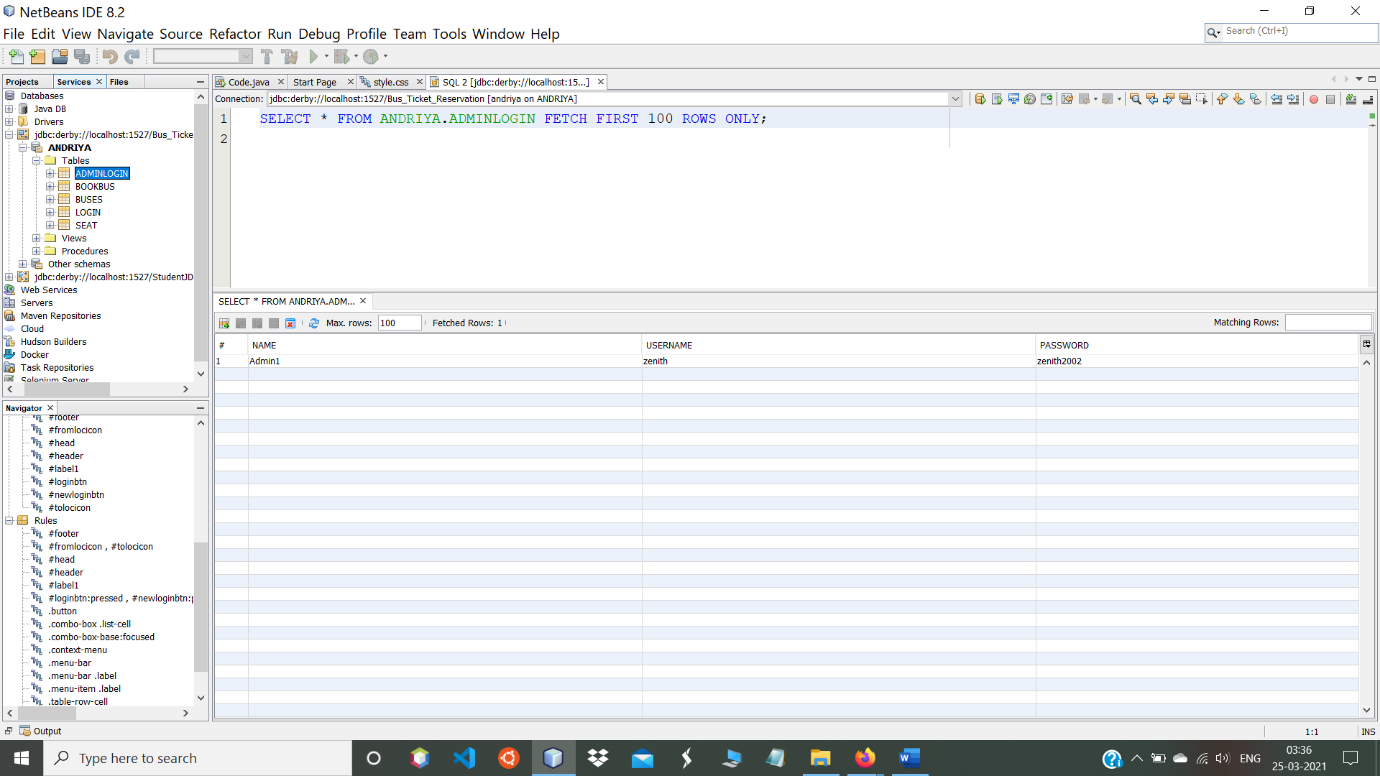
****

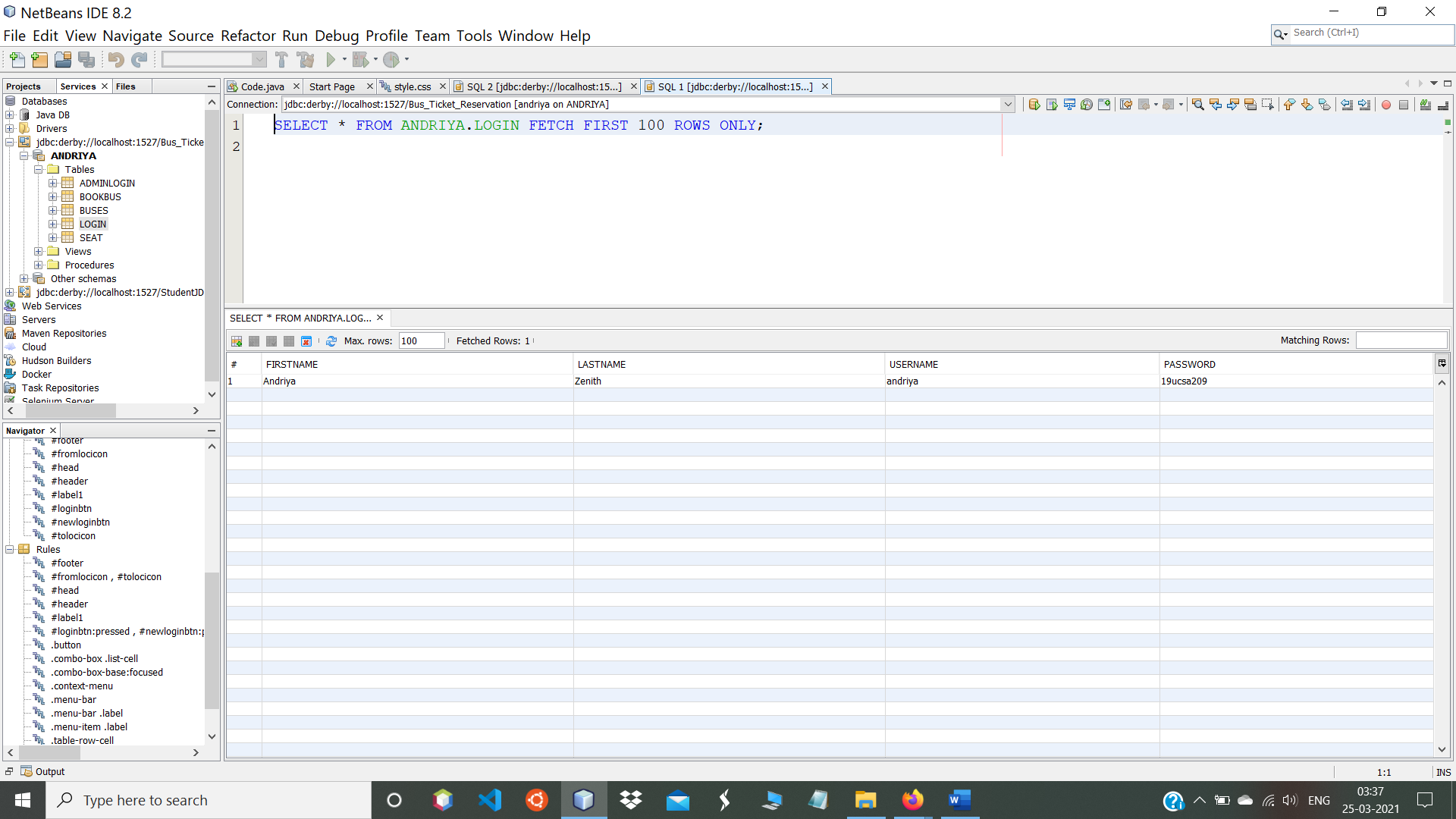
****

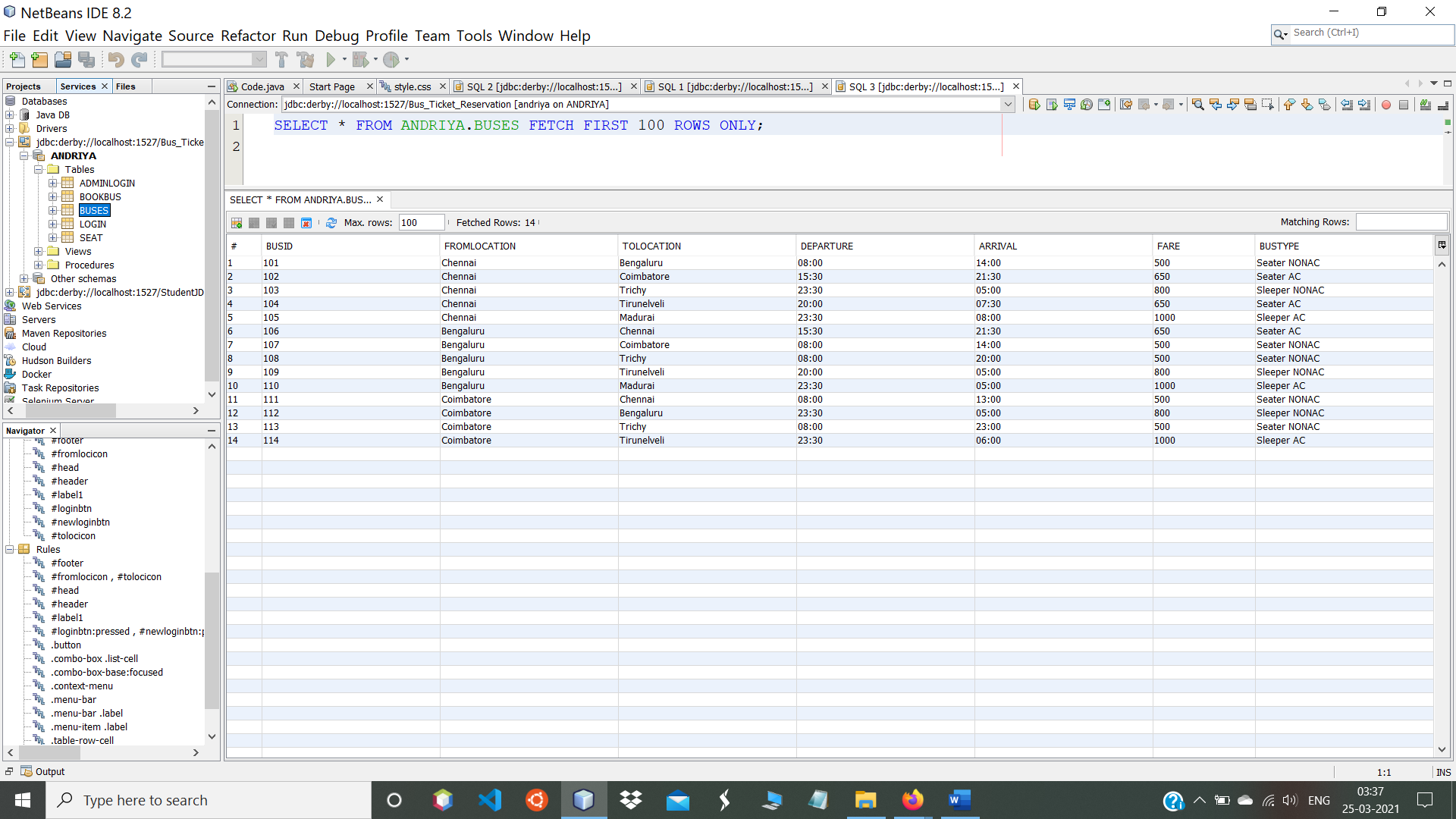
****

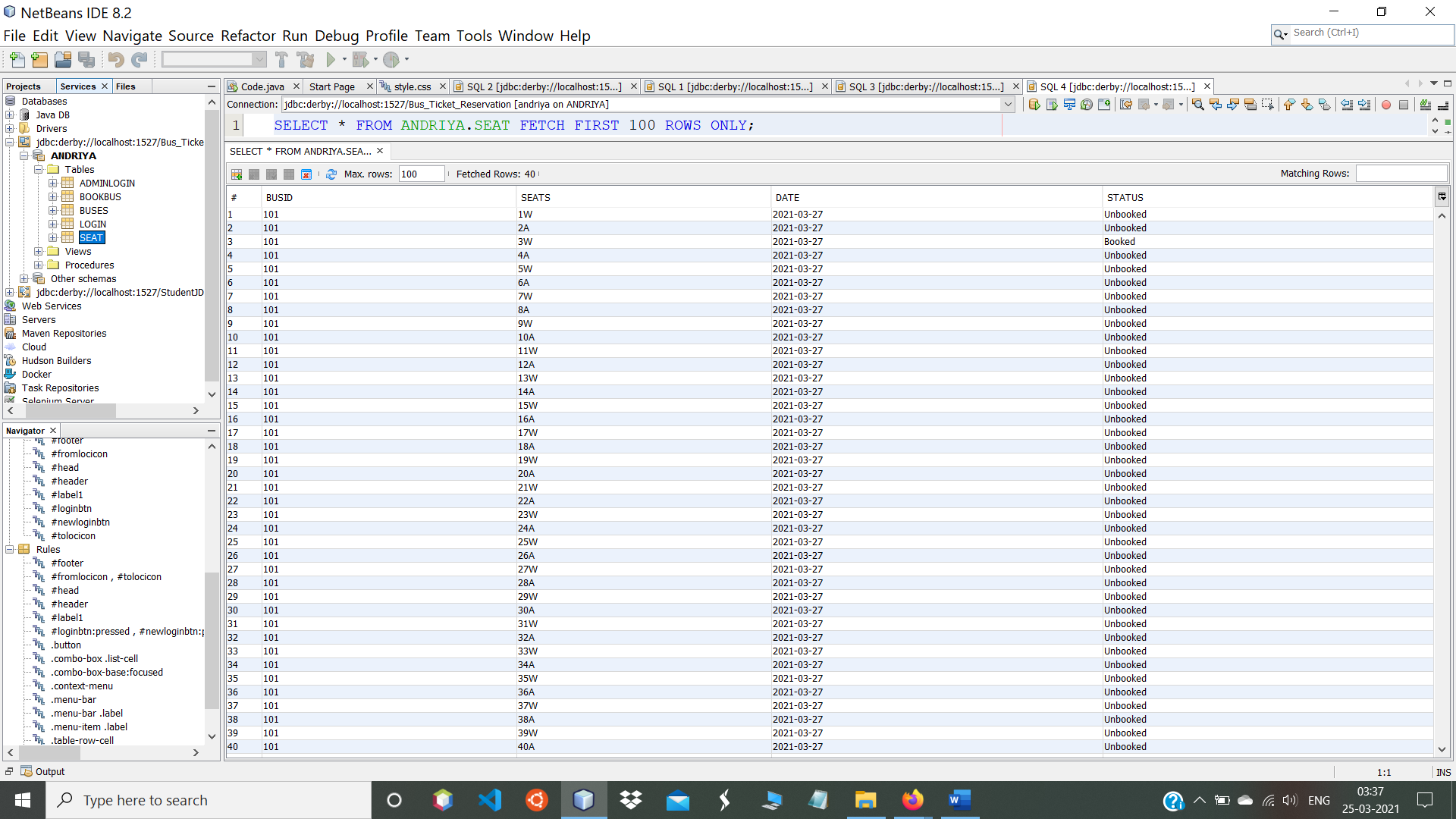
****

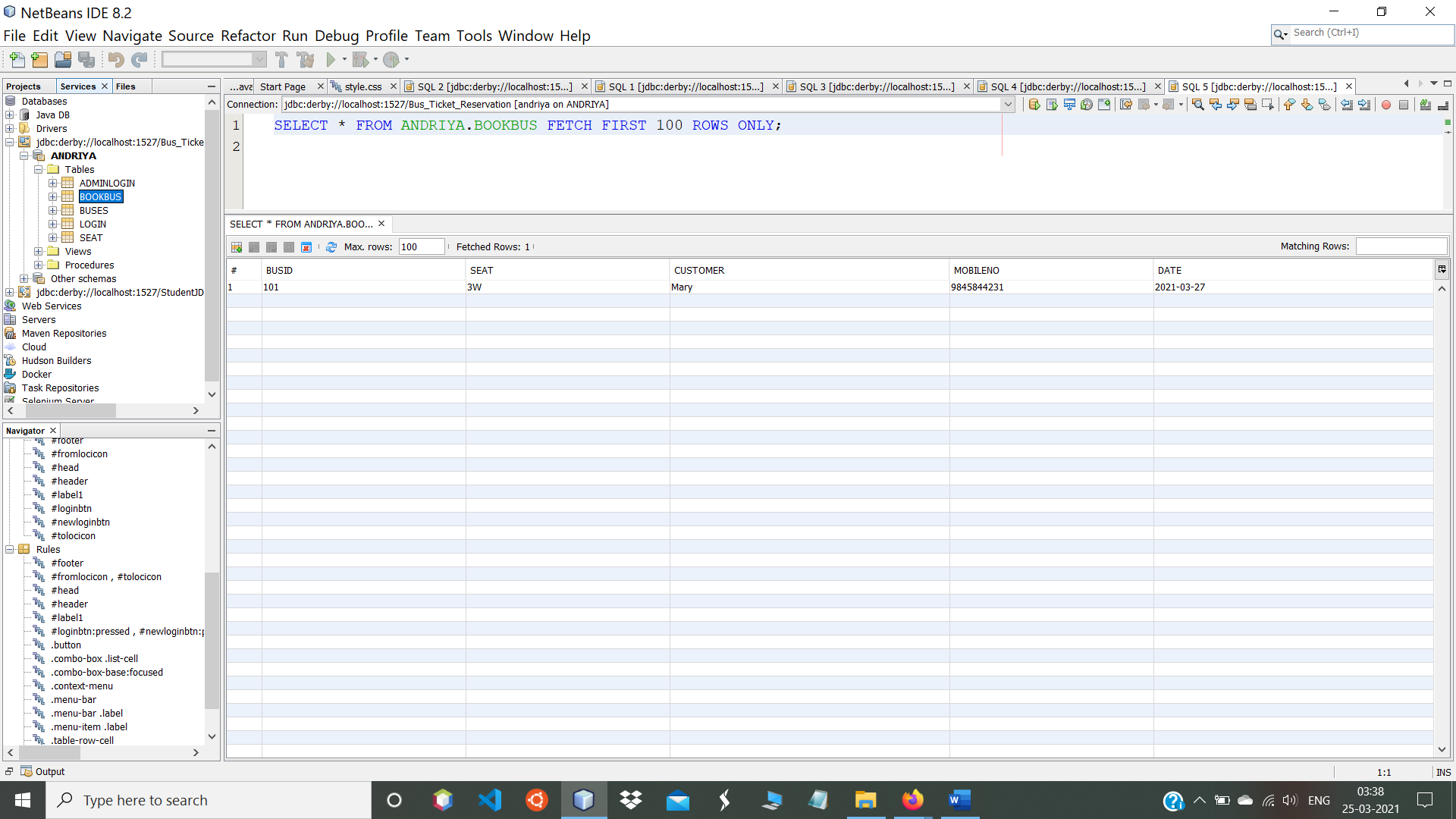
****

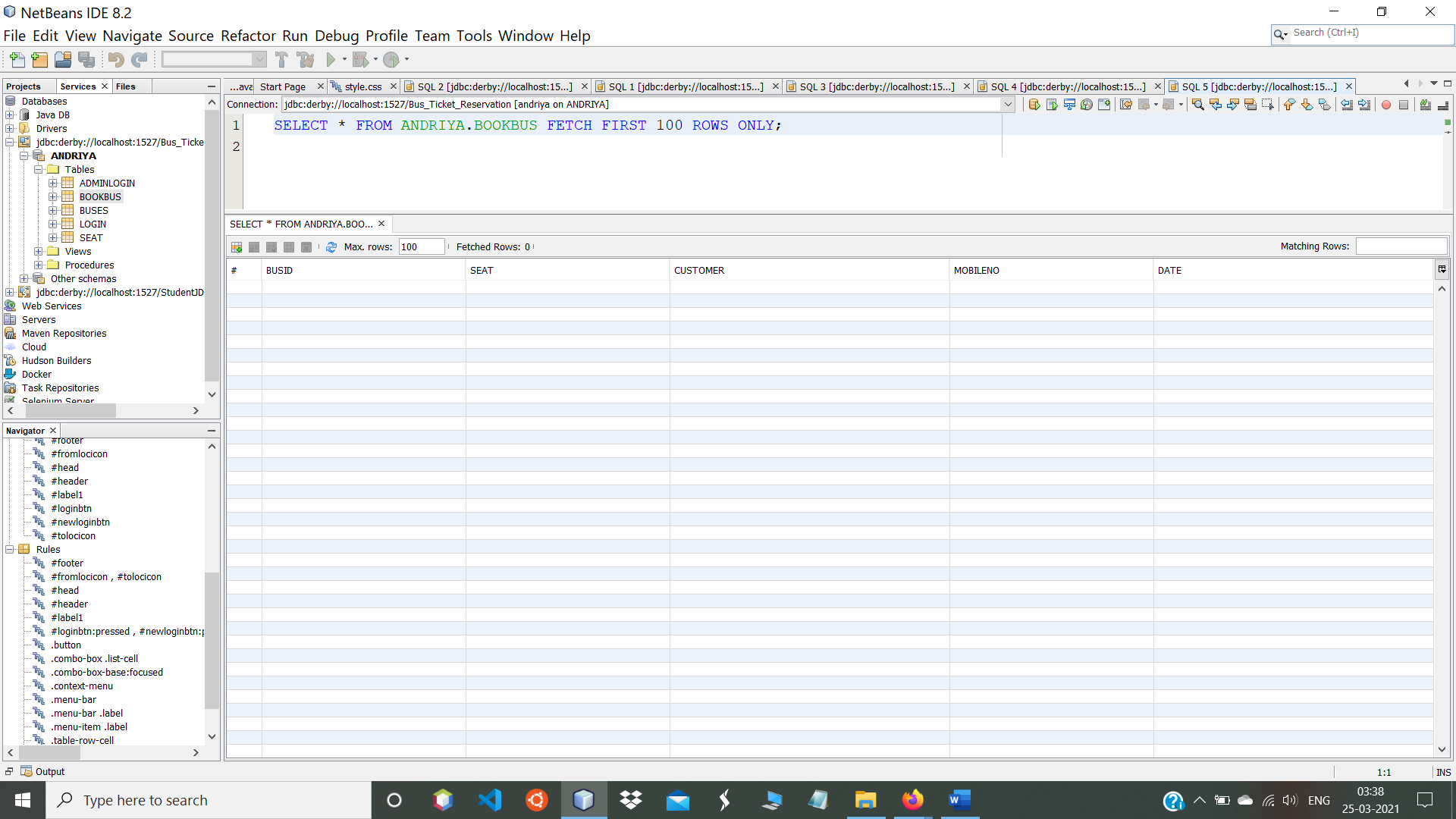
****

****

****

****

****

**s**