**Chapter 4:**

**SYSTEM IMPLEMENTATION**

**4.1 Tools and languages used:**

**4.1.1 Tools used:**

1. IntelliJ IDEA IDE Community Edition 2018.2 IDE
2. Java Development Kit 10.0.2
3. Scenebuilder 2.0
4. MySQL 8.0 Client and Shell

**4.1.2 Languages used:**

1. Java
2. JavaFX Framework
3. SQL (Structured Query Language)

**4.2 Source Code**

**4.2.1 Establishing Connection to MySQL Database**

public static Connection getConnection() throws SQLException {

String dbName="restaurantdb";

String username="root";

String password="toor";

Connection connection= DriverManager.getConnection("jdbc:mysql://localhost/"+dbName,username,password);

return connection;

}

}

**4.2.2 Validation of User**

public void login(ActionEvent actionEvent) throws SQLException, ClassNotFoundException, IOException {

Connection con=DBConnection.getConnection();

String sql="select \* from users where username='"+uname.getText()+"' and password='"+pass.getText()+"'";

PreparedStatement ps=con.prepareStatement(sql);

ResultSet rs= ps.executeQuery();

if (rs.next())

{

Stage stage = (Stage) login\_btn.getScene().getWindow();

AnchorPane root;

root = (AnchorPane) FXMLLoader.load(getClass().getResource("dashboard.fxml"));

Scene scene = new Scene(root);

stage.setScene(scene);

}

else

{

Alert errorAlert = new Alert(Alert.AlertType.ERROR);

errorAlert.setHeaderText("Authorisation Error!");

errorAlert.setContentText("Invalid Username/Password");

errorAlert.showAndWait();

}

}

}

**4.2.3 Adding Dish**

public void add\_dish(ActionEvent actionEvent) throws SQLException {

Connection con=DBConnection.getConnection();

If(dish\_name.getText(

{

Alert errorAlert = new Alert(Alert.AlertType.ERROR);

errorAlert.setHeaderText("Input Error!");

errorAlert.setContentText("Please Enter Dish Name,Category and Price");

errorAlert.showAndWait();

}

else {

String check = "select \* from dishes where dish\_name='" + dish\_name.getText() + "'";

ps = con.prepareStatement(check);

rs = ps.executeQuery();

if (rs.next()) {

Alert errorAlert = new Alert(Alert.AlertType.ERROR);

errorAlert.setHeaderText("Data Error!");

errorAlert.setContentText("Dish Already Exists");

errorAlert.showAndWait();

} else {

String sql1="insert into dishes(dish\_name,cat\_id,cat\_name) values('"+dish\_name.getText()+"',(select cat\_id from categories where cat\_name='"+dish\_cat.getValue()+"'),'"+dish\_cat.getValue()+"')";

String sql2="insert into prices values((select dish\_id from dishes where dish\_name='"+dish\_name.getText()+"'),"+dish\_price.getText()+")";

ps = con.prepareStatement(sql1);

ps.executeUpdate();

ps = con.prepareStatement(sql2);

ps.executeUpdate();

Alert infoAlert = new Alert(Alert.AlertType.INFORMATION);

infoAlert.setHeaderText("Alert!");

infoAlert.setContentText("Dish Added Successfully!");

infoAlert.showAndWait();

dish\_name.setText("");

dish\_cat.setValue("");

dish\_price.setText("");

}

}

}

}

**4.2.4 Deleting Dish**

public void delete\_dish(ActionEvent actionEvent) throws SQLException {

if(dish\_id.getText().isEmpty())

{

Alert errorAlert = new Alert(Alert.AlertType.ERROR);

errorAlert.setHeaderText("Input Error!");

errorAlert.setContentText("Please Enter Dish ID");

errorAlert.showAndWait();

}

else {

String check = "select \* from dishes where dish\_id='" + dish\_id.getText() + "'";

ps = con.prepareStatement(check);

rs = ps.executeQuery();

if (rs.next()) {

String sql = "delete from dishes where dish\_id=" + dish\_id.getText();

ps = con.prepareStatement(sql);

ps.executeUpdate();

Alert infoAlert = new Alert(Alert.AlertType.INFORMATION);

infoAlert.setHeaderText("Alert!");

infoAlert.setContentText("Dish Deleted Successfully!");

infoAlert.showAndWait();

dish\_id.setText("");

dish\_name.setText("");

dish\_cat.setText("");

dish\_price.setText("");

} else {

Alert errorAlert = new Alert(Alert.AlertType.ERROR);

errorAlert.setHeaderText("Data Error!");

errorAlert.setContentText("Invalid Dish ID!");

errorAlert.showAndWait();

}

}

}

**4.2.5 Updating Dish Details**

public void update\_dish(ActionEvent actionEvent) throws SQLException {

if(dish\_id.getText().isEmpty() || dish\_name.getText().isEmpty() || dish\_cat.getValue().isEmpty() || dish\_price.getText().isEmpty())

{

Alert errorAlert = new Alert(Alert.AlertType.ERROR);

errorAlert.setHeaderText("Input Error!");

errorAlert.setContentText("Please Enter Product ID,Name,Category and Price");

errorAlert.showAndWait();

}

else {

String sql1="update dishes set dish\_name='"+dish\_name.getText()+"',cat\_id=(select cat\_id from categories where cat\_name='"+dish\_cat.getValue()+"'),cat\_name='"+dish\_cat.getValue()+"' where dish\_id="+dish\_id.getText();

ps=con.prepareStatement(sql1);

ps.executeUpdate();

String sql2="update prices set dish\_id="+dish\_id.getText()+",price="+dish\_price.getText()+" where dish\_id="+dish\_id.getText();

ps=con.prepareStatement(sql2);

ps.executeUpdate();

Alert errorAlert = new Alert(Alert.AlertType.ERROR);

errorAlert.setHeaderText("Alert!");

errorAlert.setContentText("Dish Updated Successfully!");

errorAlert.showAndWait();

dish\_name.setText("");

dish\_cat.setValue("");

dish\_price.setText("");

dish\_id.setText("");

}

}