**Java coding challenge Documentation**

**5.Loan Management System**

**SQL PART**

CREATE DATABASE LoanManagementSystem;

USE LoanManagementSystem;

CREATE TABLE Customer (

customerId INT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phoneNumber VARCHAR(15),

address VARCHAR(255),

creditScore INT NOT NULL

);

desc Customer;

CREATE TABLE Loan (

loanId INT PRIMARY KEY,

customerId INT,

principalAmount DECIMAL(15,2) NOT NULL,

interestRate DECIMAL(5,2) NOT NULL,

loanTerm INT NOT NULL,

loanType VARCHAR(20),

loanStatus VARCHAR(20),

FOREIGN KEY (customerId) REFERENCES Customer(customerId)

);

CREATE TABLE HomeLoan (

loanId INT PRIMARY KEY,

propertyAddress VARCHAR(255),

propertyValue INT,

FOREIGN KEY (loanId) REFERENCES Loan(loanId)

);

CREATE TABLE CarLoan (

loanId INT PRIMARY KEY,

carModel VARCHAR(100),

carValue INT,

FOREIGN KEY (loanId) REFERENCES Loan(loanId)

);

INSERT INTO Customer VALUES

(1, 'Arun Kumar', 'arun.kumar@gmail.com', '9876543210', 'Chennai, TN', 720),

(2, 'Meena R', 'meena.r@yahoo.com', '9845098765', 'Coimbatore, TN', 680),

(3, 'Ravi Shankar', 'ravi.s@gmail.com', '9945012345', 'Madurai, TN', 710),

(4, 'Divya Lakshmi', 'divya.lk@yahoo.in', '9566234567', 'Salem, TN', 640),

(5, 'Vignesh V', 'vigneshv@gmail.com', '9845111222', 'Tirunelveli, TN', 780),

(6, 'Priya Dharshini', 'priya.dharshini@gmail.com', '9003287765', 'Trichy, TN', 700),

(7, 'Sathish K', 'sathish.k@gmail.com', '9798098770', 'Erode, TN', 600),

(8, 'Lakshmi Narayanan', 'lakshmi.n@gmail.com', '9823421098', 'Kanchipuram, TN', 675),

(9, 'Harini S', 'harini.s@gmail.com', '9090012345', 'Thanjavur, TN', 740),

(10, 'Mohan Raj', 'mohan.raj@gmail.com', '9654210900', 'Vellore, TN', 690);

INSERT INTO Loan VALUES

(201, 1, 500000.00, 7.5, 60, 'HomeLoan', 'Pending'),

(202, 2, 300000.00, 8.2, 48, 'CarLoan', 'Pending'),

(203, 3, 600000.00, 6.9, 72, 'HomeLoan', 'Pending'),

(204, 4, 250000.00, 9.0, 36, 'CarLoan', 'Pending'),

(205, 5, 450000.00, 7.0, 60, 'HomeLoan', 'Pending'),

(206, 6, 350000.00, 8.0, 48, 'CarLoan', 'Pending'),

(207, 7, 500000.00, 7.2, 60, 'HomeLoan', 'Pending'),

(208, 8, 280000.00, 8.5, 36, 'CarLoan', 'Pending'),

(209, 9, 750000.00, 6.5, 84, 'HomeLoan', 'Pending'),

(210, 10, 320000.00, 8.1, 48, 'CarLoan', 'Pending');

INSERT INTO HomeLoan VALUES

(201, 'No.12, Anna Nagar, Chennai', 520000),

(203, '7th Street, KK Nagar, Madurai', 630000),

(205, '14/3 Mettur Road, Tirunelveli', 470000),

(207, 'Opp. Railway Station, Salem', 510000),

(209, '17A Srirangam Main Road, Thanjavur', 790000);

INSERT INTO CarLoan VALUES

(202, 'Hyundai i20', 320000),

(204, 'Tata Nexon', 260000),

(206, 'Maruti Swift Dzire', 370000),

(208, 'Honda Amaze', 300000),

(210, 'Renault Kwid', 350000);

**Java part**

**Db.properties**

user=root

password=ruba

protocol=jdbc:mysql:

system=localhost

database=LoanManagementSystem

port=3306

**Entity package**

**Customer entity**

**package** com.loan.entity;

**public** **class** Customer {

**private** **int** customerId;

**private** String name;

**private** String email;

**private** String phoneNumber;

**private** String address;

**private** **int** creditScore;

**public** Customer() {}

**public** Customer(**int** customerId, String name, String email, String phoneNumber, String address,

**int** creditScore) {

**super**();

**this**.customerId = customerId;

**this**.name = name;

**this**.email = email;

**this**.phoneNumber = phoneNumber;

**this**.address = address;

**this**.creditScore = creditScore;

}

**public** **int** getCustomerId() {

**return** customerId;

}

**public** **void** setCustomerId(**int** customerId) {

**this**.customerId = customerId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getPhoneNumber() {

**return** phoneNumber;

}

**public** **void** setPhoneNumber(String phoneNumber) {

**this**.phoneNumber = phoneNumber;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

**public** **int** getCreditScore() {

**return** creditScore;

}

**public** **void** setCreditScore(**int** creditScore) {

**this**.creditScore = creditScore;

}

@Override

**public** String toString() {

**return** "Customer [customerId=" + customerId + ", name=" + name + ", email=" + email +

", phoneNumber=" + phoneNumber + ", address=" + address + ", creditScore=" + creditScore + "]";

}

}

**Loan entity**

**package** com.loan.entity;

**public** **class** Loan {

**private** **int** loanId;

**private** Customer customer;

**private** **double** principalAmount;

**private** **double** interestRate;

**private** **int** loanTerm;

**private** String loanType;

**private** String loanStatus;

**public** Loan() {}

**public** Loan(**int** loanId, Customer customer, **double** principalAmount, **double** interestRate, **int** loanTerm,

String loanType, String loanStatus) {

**super**();

**this**.loanId = loanId;

**this**.customer = customer;

**this**.principalAmount = principalAmount;

**this**.interestRate = interestRate;

**this**.loanTerm = loanTerm;

**this**.loanType = loanType;

**this**.loanStatus = loanStatus;

}

**public** **int** getLoanId() {

**return** loanId;

}

**public** **void** setLoanId(**int** loanId) {

**this**.loanId = loanId;

}

**public** Customer getCustomer() {

**return** customer;

}

**public** **void** setCustomer(Customer customer) {

**this**.customer = customer;

}

**public** **double** getPrincipalAmount() {

**return** principalAmount;

}

**public** **void** setPrincipalAmount(**double** principalAmount) {

**this**.principalAmount = principalAmount;

}

**public** **double** getInterestRate() {

**return** interestRate;

}

**public** **void** setInterestRate(**double** interestRate) {

**this**.interestRate = interestRate;

}

**public** **int** getLoanTerm() {

**return** loanTerm;

}

**public** **void** setLoanTerm(**int** loanTerm) {

**this**.loanTerm = loanTerm;

}

**public** String getLoanType() {

**return** loanType;

}

**public** **void** setLoanType(String loanType) {

**this**.loanType = loanType;

}

**public** String getLoanStatus() {

**return** loanStatus;

}

**public** **void** setLoanStatus(String loanStatus) {

**this**.loanStatus = loanStatus;

}

@Override

**public** String toString() {

**return** "Loan [loanId=" + loanId + ", customer=" + customer + ", principalAmount=" + principalAmount +

", interestRate=" + interestRate + ", loanTerm=" + loanTerm +

", loanType=" + loanType + ", loanStatus=" + loanStatus + "]";

}

}

**Car loan**

**package** com.loan.entity;

**public** **class** CarLoan **extends** Loan{

**private** String carModel;

**private** **int** carValue;

**public** CarLoan() {}

**public** CarLoan(**int** loanId, Customer customer, **double** principalAmount, **double** interestRate,

**int** loanTerm, String loanType, String loanStatus,String carModel, **int** carValue) {

**super**(loanId, customer, principalAmount, interestRate, loanTerm, loanType, loanStatus);

**this**.carModel = carModel;

**this**.carValue = carValue;

}

**public** String getCarModel() {

**return** carModel;

}

**public** **void** setCarModel(String carModel) {

**this**.carModel = carModel;

}

**public** **int** getCarValue() {

**return** carValue;

}

**public** **void** setCarValue(**int** carValue) {

**this**.carValue = carValue;

}

@Override

**public** String toString() {

**return** **super**.toString() + ", CarLoan [carModel=" + carModel + ", carValue=" + carValue + "]";

}

}

**HOME LOAN**

**package** com.loan.entity;

**public** **class** HomeLoan **extends** Loan {

**private** String propertyAddress;

**private** **int** propertyValue;

**public** HomeLoan() {}

**public** HomeLoan(**int** loanId, Customer customer, **double** principalAmount, **double** interestRate,

**int** loanTerm, String loanType, String loanStatus,

String propertyAddress, **int** propertyValue) {

**super**(loanId, customer, principalAmount, interestRate, loanTerm, loanType, loanStatus);

**this**.propertyAddress = propertyAddress;

**this**.propertyValue = propertyValue;

}

**public** String getPropertyAddress() {

**return** propertyAddress;

}

**public** **void** setPropertyAddress(String propertyAddress) {

**this**.propertyAddress = propertyAddress;

}

**public** **int** getPropertyValue() {

**return** propertyValue;

}

**public** **void** setPropertyValue(**int** propertyValue) {

**this**.propertyValue = propertyValue;

}

@Override

**public** String toString() {

**return** **super**.toString() + ", HomeLoan [propertyAddress=" + propertyAddress +

", propertyValue=" + propertyValue + "]";

}

}

**DAO PACKAGE**

**ILoanRepository**

**package** com.loan.dao;

**import** com.loan.entity.Loan;

**import** com.loan.exception.InvalidLoanException;

**import** java.util.List;

**public** **interface** ILoanRepository {

**void** applyLoan(Loan loan);

**double** calculateInterest(**int** loanId) **throws** InvalidLoanException;

**double** calculateInterest(**double** principal, **double** rate, **int** term);

String loanStatus(**int** loanId);

**double** calculateEMI(**int** loanId) **throws** InvalidLoanException;

**double** calculateEMI(**double** principal, **double** rate, **int** term);

**void** loanRepayment(**int** loanId, **double** amount);

List<Loan> getAllLoan();

Loan getLoanById(**int** loanId) **throws** InvalidLoanException;

}

ILoanRepositoryImpl

**package** com.loan.dao;

**import** java.util.\*;

**import** com.loan.entity.\*;

**import** com.loan.exception.InvalidLoanException;

**import** com.loan.util.DbConUtil;

**import** java.sql.\*;

**public** **class** ILoanRepositoryImpl **implements** ILoanRepository {

Scanner sc = **new** Scanner(System.***in***);

@Override

**public** **void** applyLoan(Loan loan) {

System.***out***.print("Do you want to apply for this loan? (Yes/No): ");

String confirm = sc.nextLine();

**if** (!confirm.equalsIgnoreCase("Yes")) {

System.***out***.println("Loan application cancelled by user.");

**return**;

}

Connection conn = DbConUtil.*getDbConnection*();

PreparedStatement psLoan = **null**, psSub = **null**;

**try** {

String loanQuery = "INSERT INTO Loan (loanId, loanType, customerName, principalAmount, interestRate, loanTerm, creditScore, status) " +

"VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

psLoan = conn.prepareStatement(loanQuery);

psLoan.setInt(1, loan.getLoanId());

psLoan.setString(2, loan.getLoanType());

psLoan.setString(3, loan.getCustomer().getName());

psLoan.setDouble(4, loan.getPrincipalAmount());

psLoan.setDouble(5, loan.getInterestRate());

psLoan.setInt(6, loan.getLoanTerm());

psLoan.setInt(7, loan.getCustomer().getCreditScore());

psLoan.setString(8, loan.getLoanStatus());

**int** rows = psLoan.executeUpdate();

**if** (loan **instanceof** CarLoan) {

CarLoan cl = (CarLoan) loan;

String carQuery = "INSERT INTO CarLoan (loanId, carModel, carPrice) VALUES (?, ?, ?)";

psSub = conn.prepareStatement(carQuery);

psSub.setInt(1, cl.getLoanId());

psSub.setString(2, cl.getCarModel());

psSub.setDouble(3, cl.getCarValue());

} **else** **if** (loan **instanceof** HomeLoan) {

HomeLoan hl = (HomeLoan) loan;

String homeQuery = "INSERT INTO HomeLoan (loanId, propertyLocation, propertyValue) VALUES (?, ?, ?)";

psSub = conn.prepareStatement(homeQuery);

psSub.setInt(1, hl.getLoanId());

psSub.setString(2, hl.getPropertyAddress());

psSub.setDouble(3, hl.getPropertyValue());

}

**if** (psSub != **null**) {

psSub.executeUpdate();

}

System.***out***.println("Loan application submitted successfully!");

} **catch** (SQLException e) {

System.***out***.println("Error applying loan: " + e.getMessage());

} **finally** {

**try** {

**if** (psLoan != **null**) psLoan.close();

**if** (psSub != **null**) psSub.close();

**if** (conn != **null**) conn.close();

} **catch** (SQLException ignored) {}

}

}

@Override

**public** **double** calculateInterest(**int** loanId) **throws** InvalidLoanException {

Connection conn = DbConUtil.*getDbConnection*();

PreparedStatement ps = **null**;

ResultSet rs = **null**;

**double** interest = 0.0;

**try** {

String sql = "SELECT principalAmount, interestRate, loanTerm FROM Loan WHERE loanId = ?";

ps = conn.prepareStatement(sql);

ps.setInt(1, loanId);

rs = ps.executeQuery();

**if** (rs.next()) {

**double** principal = rs.getDouble("principalAmount");

**double** rate = rs.getDouble("interestRate");

**int** term = rs.getInt("loanTerm");

interest = calculateInterest(principal, rate, term);

} **else** {

**throw** **new** InvalidLoanException("Loan with ID " + loanId + " not found.");

}

} **catch** (SQLException e) {

e.printStackTrace();

**throw** **new** InvalidLoanException("Error calculating interest: " + e.getMessage());

} **finally** {

**try** {

**if** (rs != **null**) rs.close();

**if** (ps != **null**) ps.close();

**if** (conn != **null**) conn.close();

} **catch** (SQLException ignored) {}

}

**return** interest;

}

@Override

**public** **double** calculateInterest(**double** principal, **double** rate, **int** term) {

**return** (principal \* rate \* term) / 12 / 100;

}

@Override

**public** String loanStatus(**int** loanId) {

Connection conn = DbConUtil.*getDbConnection*();

PreparedStatement psSelect = **null**;

PreparedStatement psUpdate = **null**;

ResultSet rs = **null**;

String resultMessage = "";

**try** {

String selectQuery = "SELECT creditScore FROM Loan WHERE loanId = ?";

psSelect = conn.prepareStatement(selectQuery);

psSelect.setInt(1, loanId);

rs = psSelect.executeQuery();

**if** (rs.next()) {

**int** creditScore = rs.getInt("creditScore");

String status;

**if** (creditScore > 650) {

status = "Approved";

resultMessage = "Loan ID " + loanId + " is Approved.";

} **else** {

status = "Rejected";

resultMessage = "Loan ID " + loanId + " is Rejected due to low credit score.";

}

String updateQuery = "UPDATE Loan SET status = ? WHERE loanId = ?";

psUpdate = conn.prepareStatement(updateQuery);

psUpdate.setString(1, status);

psUpdate.setInt(2, loanId);

psUpdate.executeUpdate();

} **else** {

resultMessage = "Loan ID " + loanId + " not found.";

}

} **catch** (SQLException e) {

e.printStackTrace();

resultMessage = "Error occurred while checking loan status.";

} **finally** {

**try** {

**if** (rs != **null**) rs.close();

**if** (psSelect != **null**) psSelect.close();

**if** (psUpdate != **null**) psUpdate.close();

**if** (conn != **null**) conn.close();

} **catch** (SQLException ignored) {}

}

**return** resultMessage;

}

@Override

**public** **double** calculateEMI(**int** loanId) **throws** InvalidLoanException {

Connection conn = DbConUtil.*getDbConnection*();

PreparedStatement ps = **null**;

ResultSet rs = **null**;

**double** emi = 0.0;

**try** {

String sql = "SELECT principalAmount, interestRate, loanTerm FROM Loan WHERE loanId = ?";

ps = conn.prepareStatement(sql);

ps.setInt(1, loanId);

rs = ps.executeQuery();

**if** (rs.next()) {

**double** principal = rs.getDouble("principalAmount");

**double** annualRate = rs.getDouble("interestRate");

**int** term = rs.getInt("loanTerm");

emi = calculateEMI(principal, annualRate, term);

} **else** {

**throw** **new** InvalidLoanException("Loan with ID " + loanId + " not found.");

}

} **catch** (SQLException e) {

e.printStackTrace();

**throw** **new** InvalidLoanException("Error while calculating EMI: " + e.getMessage());

} **finally** {

**try** {

**if** (rs != **null**) rs.close();

**if** (ps != **null**) ps.close();

**if** (conn != **null**) conn.close();

} **catch** (SQLException ignored) {}

}

**return** emi;

}

@Override

**public** **double** calculateEMI(**double** principal, **double** annualRate, **int** termMonths) {

**double** monthlyRate = annualRate / 12 / 100;

**if** (monthlyRate == 0) {

**return** principal / termMonths; // No interest case

}

**double** numerator = principal \* monthlyRate \* Math.*pow*(1 + monthlyRate, termMonths);

**double** denominator = Math.*pow*(1 + monthlyRate, termMonths) - 1;

**return** numerator / denominator;

}

@Override

**public** **void** loanRepayment(**int** loanId, **double** amount) {

Connection conn = DbConUtil.*getDbConnection*();

PreparedStatement psSelect = **null**;

PreparedStatement psUpdate = **null**;

ResultSet rs = **null**;

**try** {

String sql = "SELECT principalAmount, interestRate, loanTerm FROM Loan WHERE loanId = ?";

psSelect = conn.prepareStatement(sql);

psSelect.setInt(1, loanId);

rs = psSelect.executeQuery();

**if** (rs.next()) {

**double** principal = rs.getDouble("principalAmount");

**double** rate = rs.getDouble("interestRate");

**int** term = rs.getInt("loanTerm");

**double** emi = calculateEMI(principal, rate, term);

**if** (amount < emi) {

System.***out***.println("Repayment amount is less than 1 EMI. Payment rejected.");

**return**;

}

**int** paidEmis = (**int**) (amount / emi);

**int** remainingTerm = term - paidEmis;

**if** (remainingTerm < 0) remainingTerm = 0;

String updateQuery = "UPDATE Loan SET loanTerm = ? WHERE loanId = ?";

psUpdate = conn.prepareStatement(updateQuery);

psUpdate.setInt(1, remainingTerm);

psUpdate.setInt(2, loanId);

psUpdate.executeUpdate();

System.***out***.println("Payment accepted. EMIs paid: " + paidEmis +

", Remaining EMIs: " + remainingTerm);

} **else** {

System.***out***.println("Loan with ID " + loanId + " not found.");

}

} **catch** (SQLException e) {

e.printStackTrace();

System.***out***.println("Error during loan repayment.");

} **finally** {

**try** {

**if** (rs != **null**) rs.close();

**if** (psSelect != **null**) psSelect.close();

**if** (psUpdate != **null**) psUpdate.close();

**if** (conn != **null**) conn.close();

} **catch** (SQLException ignored) {}

}

}

@Override

**public** List<Loan> getAllLoan() {

List<Loan> loanList = **new** ArrayList<>();

Connection conn = DbConUtil.*getDbConnection*();

PreparedStatement psLoan = **null**, psSub = **null**;

ResultSet rsLoan = **null**, rsSub = **null**;

**try** {

String loanQuery = "SELECT \* FROM Loan";

psLoan = conn.prepareStatement(loanQuery);

rsLoan = psLoan.executeQuery();

**while** (rsLoan.next()) {

**int** loanId = rsLoan.getInt("loanId");

String loanType = rsLoan.getString("loanType");

String customerName = rsLoan.getString("customerName");

**double** principal = rsLoan.getDouble("principalAmount");

**double** rate = rsLoan.getDouble("interestRate");

**int** term = rsLoan.getInt("loanTerm");

**int** creditScore = rsLoan.getInt("creditScore");

String status = rsLoan.getString("status");

// Create customer

Customer customer = **new** Customer();

customer.setName(customerName);

customer.setCreditScore(creditScore);

Loan loan = **null**;

**if** (loanType.equalsIgnoreCase("Car")) {

String carQuery = "SELECT \* FROM CarLoan WHERE loanId = ?";

psSub = conn.prepareStatement(carQuery);

psSub.setInt(1, loanId);

rsSub = psSub.executeQuery();

**if** (rsSub.next()) {

String model = rsSub.getString("carModel");

**int** price = rsSub.getInt("carPrice");

loan = **new** CarLoan(loanId, customer, principal, rate, term, loanType, status, model, price);

}

} **else** **if** (loanType.equalsIgnoreCase("Home")) {

String homeQuery = "SELECT \* FROM HomeLoan WHERE loanId = ?";

psSub = conn.prepareStatement(homeQuery);

psSub.setInt(1, loanId);

rsSub = psSub.executeQuery();

**if** (rsSub.next()) {

String location = rsSub.getString("propertyLocation");

**int** value = rsSub.getInt("propertyValue");

loan = **new** HomeLoan(loanId, customer, principal, rate, term, loanType, status, location, value);

}

}

**if** (loan != **null**) {

loanList.add(loan);

System.***out***.println(loan); // Print loan details

}

}

} **catch** (SQLException e) {

e.printStackTrace();

System.***out***.println("Error fetching loans.");

} **finally** {

**try** {

**if** (rsLoan != **null**) rsLoan.close();

**if** (rsSub != **null**) rsSub.close();

**if** (psLoan != **null**) psLoan.close();

**if** (psSub != **null**) psSub.close();

**if** (conn != **null**) conn.close();

} **catch** (SQLException ignored) {}

}

**return** loanList;

}

@Override

**public** Loan getLoanById(**int** loanId) **throws** InvalidLoanException {

Connection conn = **null**;

PreparedStatement ps = **null**;

ResultSet rs = **null**;

Loan loan = **null**;

**try** {

conn = DbConUtil.*getDbConnection*();

String query = "SELECT \* FROM Loan WHERE loanId = ?";

ps = conn.prepareStatement(query);

ps.setInt(1, loanId);

rs = ps.executeQuery();

**if** (rs.next()) {

loan = **new** Loan();

loan.setLoanId(rs.getInt("loanId"));

loan.setLoanType(rs.getString("loanType"));

loan.setPrincipalAmount(rs.getDouble("principalAmount"));

loan.setInterestRate(rs.getDouble("interestRate"));

loan.setLoanTerm(rs.getInt("loanTerm"));

loan.setLoanStatus(rs.getString("status"));

System.***out***.println("Loan Details:");

System.***out***.println("Loan ID : " + loan.getLoanId());

System.***out***.println("Loan Type : " + loan.getLoanType());

System.***out***.println("Principal : ₹" + loan.getPrincipalAmount());

System.***out***.println("Interest Rate : " + loan.getInterestRate() + "%");

System.***out***.println("Loan Term : " + loan.getLoanTerm() + " months");

System.***out***.println("Status : " + loan.getLoanStatus());

} **else** {

**throw** **new** InvalidLoanException("Loan with ID " + loanId + " not found.");

}

} **catch** (SQLException e) {

**throw** **new** InvalidLoanException("Error retrieving loan: " + e.getMessage());

} **finally** {

**try** {

**if** (rs != **null**) rs.close();

**if** (ps != **null**) ps.close();

**if** (conn != **null**) conn.close();

} **catch** (SQLException ignored) {}

}

**return** loan;

}

}

**Util package**

**DbConUtil**

**package** com.loan.util;

**import** java.io.IOException;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**public** **class** DbConUtil {

**private** **static** **final** String ***fileName***="db.properties";

**public** **static** Connection getDbConnection() {

Connection con=**null**;

String connString=**null**;

**try** {

connString=DBPropertyUtil.*getConnectionString*(***fileName***);

}**catch** (IOException e) {

System.***out***.println("Connection String Creation Failed");

e.printStackTrace();

}

**if**(connString!=**null**) {

**try** {

Class.*forName*("com.mysql.cj.jdbc.Driver");

con=DriverManager.*getConnection*(connString);

}**catch** (SQLException | ClassNotFoundException e) {

System.***out***.println("Error While Establishing DBConnection........");

e.printStackTrace();

}

}

**return** con;

}

}

**DBPropertyUtil**

**package** com.loan.util;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.util.Properties;

**public** **class** DBPropertyUtil {

**public** **static** String getConnectionString(String fileName)**throws** IOException {

//fileName="db.properties"

String connStr=**null**;

Properties props=**new** Properties();

FileInputStream fis=**new** FileInputStream(fileName);

props.load(fis);

String user=props.getProperty("user");

String password=props.getProperty("password");

String protocol=props.getProperty("protocol");

String system=props.getProperty("system");

String database=props.getProperty("database");

String port=props.getProperty("port");

connStr=protocol+"//"+system+":"+port+"/"+database+"?user="+user+"&password="+password;

System.***out***.println("DB Connection String: " + connStr);

**return** connStr;

}

}

**Exception package**

**InvalidLoanException**

**package** com.loan.exception;

**public** **class** InvalidLoanException **extends** Exception {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** InvalidLoanException(String message) {

**super**(message);

}

}

**Main package**

**package** com.loan.main;

**import** com.loan.dao.ILoanRepository;

**import** com.loan.dao.ILoanRepositoryImpl;

**import** com.loan.entity.Customer;

**import** com.loan.entity.Loan;

**import** com.loan.entity.HomeLoan;

**import** com.loan.entity.CarLoan;

**import** com.loan.exception.InvalidLoanException;

**import** java.util.Scanner;

**public** **class** LoanManagement {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

ILoanRepository repo = **new** ILoanRepositoryImpl();

**while** (**true**) {

System.***out***.println("\n=== Loan Management System ===");

System.***out***.println("1. Apply Loan");

System.***out***.println("2. Get All Loans");

System.***out***.println("3. Get Loan by ID");

System.***out***.println("4. Repay Loan");

System.***out***.println("5. Exit");

System.***out***.print("Enter your choice: ");

**int** choice = sc.nextInt();

sc.nextLine(); // consume newline

**switch** (choice) {

**case** 1:

System.***out***.print("Enter Loan ID: ");

**int** loanId = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Customer ID: ");

**int** customerId = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Customer Name: ");

String name = sc.nextLine();

System.***out***.print("Enter Credit Score: ");

**int** creditScore = sc.nextInt();

System.***out***.print("Enter Loan Type (Car/Home): ");

String type = sc.next();

sc.nextLine();

System.***out***.print("Enter Principal Amount: ");

**double** principal = sc.nextDouble();

System.***out***.print("Enter Interest Rate: ");

**double** rate = sc.nextDouble();

System.***out***.print("Enter Loan Term (in months): ");

**int** term = sc.nextInt();

sc.nextLine();

Customer customer = **new** Customer();

customer.setCustomerId(customerId);

customer.setName(name);

customer.setCreditScore(creditScore);

Loan loan = **null**;

**if** (type.equalsIgnoreCase("Car")) {

System.***out***.print("Enter Car Model: ");

String model = sc.nextLine();

System.***out***.print("Enter Car Price: ");

**int** price = sc.nextInt();

loan = **new** CarLoan(loanId, customer, principal, rate, term, "Car", "Pending", model, price);

} **else** **if** (type.equalsIgnoreCase("Home")) {

System.***out***.print("Enter Property Location: ");

String location = sc.nextLine();

System.***out***.print("Enter Property Value: ");

**int** value = sc.nextInt();

loan = **new** HomeLoan(loanId, customer, principal, rate, term, "Home", "Pending", location, value);

}

**if** (loan != **null**) {

repo.applyLoan(loan);

} **else** {

System.***out***.println("Invalid loan type.");

}

**break**;

**case** 2:

repo.getAllLoan();

**break**;

**case** 3:

System.***out***.print("Enter Loan ID: ");

**int** searchId = sc.nextInt();

**try** {

repo.getLoanById(searchId);

} **catch** (InvalidLoanException e) {

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

}

**break**;

**case** 4:

System.***out***.print("Enter Loan ID: ");

**int** repayId = sc.nextInt();

System.***out***.print("Enter repayment amount: ");

**double** amount = sc.nextDouble();

repo.loanRepayment(repayId, amount);

**break**;

**case** 5:

System.***out***.println("Exiting application...");

System.*exit*(0);

**default**:

System.***out***.println("Invalid choice. Please try again.");

sc.close();

}

}

}

}

1)









