Page No.
Date

Assignment - C2

* Date of Submission: 30/11/2020
* Date of Completion: 30/11/2020
* Title: Java - MySQL connectivity
* Problem Statements:
  Implement MySQL/Oracle database connectivity with PHP/python/Java. Implement Database navigation operations (add, delete, edits) using ODBC/JDBC.
* Objectives:
  - To understand concept of JDBC and MySQL connectivity
  - To perform basic database operations.
* Outcomes:
  Students will be able to:
  Successfully implement & study database connectivity and perform basic operations using it.
* S/W And H/W Requirements:
  MySQL, Java IntelliJ IDE, i5 processor, 8GB RAM, Windows 10, etc.

Teacher's Sign.: _____

Page No.
Date

* Theory:

o MySQL

- MySQL is the most popular open source Relational SQL database management system.
- It is one of the best RDBMS being used for developing various web-base s/w applications.

o Establishing JDBC connection in Java

What is JDBC?

- Its an acronym for Java Database connectivity. It is an advancement of ODBC (open database connectivity)
- JDBC is a standard API specification developed in order to move data from front end to back end.
- This API consists of classes and interfaces written in 'Java'.
- It basically acts an interface or channel between your Java program & a database i.e. it establishes a link between the two so that a programmer should send data from Java code & store it in the database for future use.

Teacher's Sign.:

Page No.
Date

* Steps for connecting Java program & database

1. Loading the Driver:

```
Class.forName("com.mysql.cj.jdbc.Driver")
.DriverManager.registerDriver()
```

2. Create connection:

```
Connection conn = DriverManager.getConnection(url,
                                    username, password);
url = "jdbc:mysql://localhost:3306/assignment_c2";
```

3. Create a Statement:

```
Statement stmt = conn.createStatement();
```

4. Execute query

(1) Query for retrieving data

```
stmt.executeQuery(query);
```

(2) Query for updating/inserting data:

```
stmt.executeUpdate(query);
```

5. Close the connection:

```
conn.close();
```

* Creating A Table:

Syntax:

```
String q = "create table tablename
            (id int, name varchar(10));"
```

Teacher's Sign.: _____

stmt.executeUpdate(q);

* Inserting into a table:

```
q = "insert into table_name values (1,'P');"
stmt.executeUpdate(q);
```

* Deleting From a Table:

```
q = "delete from table_name where id = 101";
stmt.executeUpdate(q);
```

* Retrieving from a table:

```
ResultSet rst;
    q = "select * from table_name";
rst = stmt.executeQuery(q);

while(rst.next()){
        print(rst.getString(0));
}
```

* Conclusion:

Thus, we implemented JDBC connectivity & performed basic navigation operations on it.

Teacher's Sign.: _____