Topic...Date..............................

Course Title : Advanced Internet Technologies

Name : Sahil   Course code : MCS-051

Assignment No : MCA(V)/051/Assign/2021-22

Enrollment No : 197393829   Maximum Marks : 100

Ques 1 : Write and run a server program for online submission of an examination form which comprizes the following fields and display its records.

- Student Name
- Enrolement Number.
- Course code (s)
- Regional center code
- Email id

You are required to create at least 5 records in a database in and connect the servlet with the database using JDBC API and run the SQL query.

```
import java.io.x;
import java.sql.x;
importx.servlet.servlet Exception;
import javax.servlet.http.*;
public void doget (Httpservlet.Response response)
 throw servlet Exception, IO Exception
{
response.set content Type ("text/html");
print writor out response.get writer ();
String name = request ("name");
String student name = request.get parameter
        ("s.name");
String enrolement number = request.get parameter
        ("enroll.number");
```

Teacher's Sign..........................

Topic...Date...............................

```
String email Id = request. get parameter ("email Id");
String grade achieved = request. get parameter
        (" grade. achiev");
Class. yourname (" oracle. jdbc. driver. oracle
        Driver");
connection on = Drivermanager. get connection
        (jdbc. oracle: thin. : @localhost : 1521 :
    Xe", "system", "oracle";
prepared statement ps - cos, prepare statement
"select * from result where enrollment number = ?";
PS. set string ()(S name);
PS. set int (enroll. number);
PS. set string (3, email ID);
PS. set string (4 grade. achiev);
out. print ("<captain> Result: </caption>");
Result set metadat. rsm. r. s. get metadata ();
out. print ("<tr>");
for (int i=1; i<= total; i++)

{
out. print ("<+n>"tremd. get columnname(1)+
 "</th>");
}
out. print ("</tr">);
while (r/. next())
{
out. print ("<tr><td>" + rs. get. string (1)"
"</td><td>"trs. get int (2)+"</td>"
"<td>" trs. get string (3)+ "</td><td>"+
rs. get string (4) + "</td></td>"+ rs. get
string (S) + </td> rs. get string (10) + "</td>
</tr>");
```

Teacher's Sign..........................

```
}
 out.print (<"/table>");
}
 catch
   (Exception 2)
}
{
  finally
        { out.close();
 }
  }
   }
```

Ques 2: What is the purpose of session tracking? Describe the methods used by servlet for session tracking.

session Tracking tracks a user's requests and maintains their state. It is a mechanism used to store information on specific users and in order to recognize these user's requests when they connect to the web server.

HTTP is a stateless protocol where each request to the server is treated like a new request. However, there are ways to maintain the data for a particular user in order to serve them specific content.

There are several ways to track user sessions including cookies, URL rewriting and hidden form fields.

Topic..Date...............................

1) public ~~[illegible]~~ object getAttribute (String name)
This method returns the object bound with the specified name in this session, or null if no object is bound under the name.

2) public Enumeration getAttributeNames()
This method returns an Enumeration of string objects containing the names of all the objects bound to this session.

3) public long getCreationTime()
This method returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.

4) public string getId()
This method returns a string containing the unique identifier assigned to this session.

5) public long getlastAccessedTime()
This method returns the last accessed time of the session, in the format of milliseconds since midnight January 1, 1970 GMT.

6) public int getMaxInactiveInterval()
This method returns the maximum time interval, that the servlet container will keep the session open b/w client access.

7) public void invalidate()
This method invalidates this session and unbinds any objects bound to it.

8) public boolean isNew(
This method returns true if the clients does not yet know about the session or if the client chooses not to join the session.

Teacher's Sign.............................

Topic...Date..............................

9, public void removeAttribute (String name)
This method removes the object bound with the specified name from this session.

10, public void setAttribute (String name, Object value)
This method binds an object to this session, using the name specified.

11, public void setMaxInactiveInternally (int interval)
This method specifies the time, in seconds, b/w client requests before the servlet container will invalidate this session.

Ques 3 : a, Describe all the JDBC SQL statements APIs and its methods.

The statement interface is used to create SQL basic statements in Java it provides methods to execute queries with the database. There are different types of statements that are used in JDBC as follows :

- Create statement
- Prepared statement
- Callable statement

1, Create a statement :
From the connection interface, you can create the Object for this interface. It is generally used for general - purpose access to databases and is useful while using static SQL statements at runtime.
Syntax :
Statement statement = connection. createStatement();
Implementation : Once the statement object is created, there are three ways to execute it.

Teacher's Sign...........................

Topic...Date.........................

- boolean execute (String SQL) : If the resultset object is retrieved, then it returns true else false is returned. Is used to execute SQL DDL statements or for dynamic SQL.
- int executeUpdate (String SQL) : Returns number of rows that are affected by the execution of the statement, used when you need a no. ofor INSERT, DELETE or UPDATE statement.
- Resultset executeQuery (String SQL) : Returns a result set object. Used similarly as select is used in SQL.

2) Prepared Statement

It represents a recompiled SQL statement, that can be executed many times. This accepts parameterized SQL queries. In this, "?" is used instead of the parameter, one can pass the parameter dynamically by using the methods of PREPARED STATEMENT at run time.

Illustration:

considering in the people database if there is a need to INSERT some values, SQL statements such as these are used:

INSERT INTO people VALUES ("Ayan", 25);

INSERT INTO people VALUES ("Kriya", 32);

Implementation : Once the PreparedStatement object is created, there are three ways to execute it :

- execute () : This return a boolean value and executes a static SQL statement that is present in the prepared statement object.

Teacher's Sign.........................

Topic.. Date..............................

- executeQuery(): Returns a resultset from the current prepared statement.
- executeUpdate(): Returns the no. of rows affected by the DML statements such as INSERT, DELETE, and more that is present in the current Prepared statement.

3) Callable statement: are stored procedures which are a group of statements that we compile in the database for some task, they are beneficial when we are dealing with multiple tables with complex scenario and rather than sending multiple queries to the database, we can send the required data to the stored procedure and lower the logic executed in the database server itself.
syntax: To prepare a Callable statement
Callable statement cstmt = con.prepareCall("{call Procedure_name(?, ?}");

Implementation: Once the callable statement object is created
- execute() is used to perform the execution of the statement.

Ques 3: b) Modify the database created in Q1 by adding 5 more record using the JDBC SQL statement method.

```
<%@page import = "java.sql.*"%>
<%!
int stu-id = 1;
String stu_name = "Harshita";
string program"= BBA"
```

Teacher's Sign..............................

Topic...Date..............................

```
String sem = "1";
String address = " Bhopal";
%>
<%
try { class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager. getConnection ("jdbc:
odbc: ignousolver");
statement st = con. createStatement ();
String query = " insert into student
values ("+stu_id ", "+stu_name + "', " + program + ",
" +sem+ ", "+address + ")";
if (st. executeUpdate (query) >0) {
out. write ("Records inserted");
} else {
out. write ("insertion faild");
}} catch ( Exception e) {
out. write (" Exception: "+e); }
%>
second Record
% @page import = "java.sql.*" %
<%!
int stu_id = 2; string stu_name = " Ishita "; String
program = " BBA"
String sem = "1";
String address = " Bhopal";
%>
< %
try {
class. forName ("sun.jdbc.odbc.JdbcDriver");
Connection con = DriverManager. get Connection
    ("jdbc: odbc: ignousolver");
statement st = con. createStatement ();
```

Teacher's Sign..........................

Topic..Date..............................

```
String query = "insert into student
 values ("+ stu_id + ",'" + stu_name + "','"+ program+ ","
 +sem+ ", "," + address + " )";
if (st. executeUpdate( query) > 0) {
out. write (" Records inserted ");
} else {
out. write (" insertion faild ");
} ~~else~~ {
~~out. write~~
} catch ( Exception e) {
out. write (" Exception ~~e~~\{ : "+e);
}
%>
```

Ques 4: Write a JSP code fragments for the following tasks :

a) Insert the following fields into a database after extracting them from an HTML from.
Student_name
Enrolmentnumber
email ID

Step 1 : Import required packages

```
import java. sql.*;
public class JDBCExample {
// JDBC driver name and database URL
static final String JDBC_DRIVER = "com. mysql. jdbc.
 Driver";
static final string DB URL = "jdbc: mysql :// local
 host / Student ";
 // Database credentials
 static final String USER = "studentname";
static final string PASS = " password";
```

Teacher's Sign..........................

Topic...Date...............................

```
public static void main ( String[] args ) {
   Connection conn = null;
 Statement stmt = null;
 try {
// Step 2: Register JDBC driver
 Class.forName ("com.mysql.jdbc.Driver");
// Step 3: Open a connection
system.out.printen (" Connecting to a selected
  database.... ");
conn = DriverManager.getConnection (DB_URL, STUDENT,
  PASS);
System.out.println (" Connected database successfully...");
//Step4 : Execute a query
system.out.printen (" Creating statement.... ");
stmt = conn.createStatement ();
// Extract records in ascending order by first name.
system.out.println (" Fetching records in ascending
  order... ");
String sql = " SELECT Student_name, Enrolment_number,
Student_mail FROM Registration "+
 " ORDER BY first ASC";
Resultset rs = Stmt.executeQuery (sql);
while (rs.next()) {
// Retrieve by column name
String Student_name = rs.getString ("name");
int Student_enrolment = rs.getInt (" enrolment ");
String Student_mail = rs.getString ( " mail ");
// Display values
System.out.print (" NAME :"+ name);
System.out.print (" Enrolment : "+ enrolment);
System.out.printen (", Mail : "+ mail);
  }
```

Teacher's Sign..............................

```
rs.close();
}
catch (SQL Exception se) {
// Handle errors for JDBC
se.printStackTrace();
} catch (Exception e) {
// Handle errors for Class.forName
e.printStackTrace();
} finally {
// finally block used to close resources
try {
if (stmt! =null)
conn.close();
} catch (SQL Exception se) {
} // do nothing
try {
if (conn! = null)
conn.close();
} catch (SQL Exception se) {
se.printStackTrace();
} // end finally try
} //end try
system.out.printen (" Goodbye!");
} // end main
} // end JDBC Example
```

b) Retrieve records from a database using JSP.
For retrieve data from MySQL database using JSP we have to create a table in database After create a table in the MySQL database you need to insert record or data on it. If you want to know how to insert data in

Topic...Date.............................

jsp please visit the link : Insert data in JSP.
The SELECT statement is used to retrieve data from one or more tables:
The SQL query for retrieve specific column
SELECT column_name(S) FROM table_name
or we can use the * character to retrieve All columns from a table:
SELECT * FROM table_name
In this example we retrieve the employee data of a company.
The JSP file for retrieving data from database is :

```
<%@page import = "java.sql.DriverManager"%>
<%@ pageimport = "java.sql.ResultSet "%>
<%@ pageimport = "java.sql.statement "%>
<%@ pageimport = "java.sql.Connection "%>
<%
String id = request.getParameter ("userid");
String driver = " com.mysql.jdbc.Driver ";
String connectionUrl = "jdbc:mysql://localhost:
 3306/";
String database = "test";
String userid = "root";
String password = "";
try {
Class.forName (driver);
} catch ( ClassNotFoundException e) {
e.printStackTrace ();
}
Connection connection = null;
Statement statement = null;
Resultset resultset = null;
```

Teacher's Sign..........................

```
%>
<!DOCTYPE html>
<html>
<body>
<h1> Retrieve Data from database in jsp </h1>
< table border = "1 ">
< tr >
< td > firstname </td>
< td > lastname </ td >
<td> City name </ td>
<td> Email </td>

</tr>
<%
try {
connection = DriverManager. getConnection ( connectionUrl +
 database, userid, password );
statement = connection . createStatement ();
String sql = "select * from users";
 resultSet = statement . executeQuery (sql);
 while (resultSet. Next ()) {
 %>
 <tr>
<td><%= resultset. getString ("first_name") %></td>
<td><%=resultset.getString ("last_name") %></td>
<td><%= resultset. getString (" city_name ") %></td>
<td><%= result Set. getString ("email ") %></td>
</tr>
<%
}
connection . close ();
} catch ( Exception e) {
```

Topic...Date...............................

```
e.printStackTrace();
}
%>
</table>
</body>
</html>
```

Ques 5: a) Write an application to create a XML document from an employee database. The XML database document should contain the name of the employee, address, mobile number and the last 6 months mobile recharge payment summary.

```
<! employee.xml ....>
<! --- Representing the information of employee in
  XML document >
<! DataType employee "syst.employee.dtd>
< information>
< name>
</ name> Singh </ name >
< mname > chandra </ mname >
< fname > Lokesh </ fname >
</ name>    <address >
< city > Lucknow </ city>
< state > UttarPradesh </ state >
< Pincode > 226012 </ pin code >
< mobile no. >
< mob. > 9836123456 </ mob.>
< Recharge >
< Jan > Rs.299 </ Jan >
< Feb. > Rs.299 </ Feb >
< March > Rs.10 </ March >
< April > Rs.30 </ April >
```

Teacher's Sign...........................

Topic...Date...............................

```
<May > Rs10 </May>
< June> Rs. 20 </ June>
 DTD  code for XML information
<? XML. version = "1.0" and = "internal"? >
<! ELEMENT employee < grade+, content *>>
<! ELEMENT   grade <name, address, mobile no.,
      recharge >
<! ELEMENT name <lname, mname, fname >
<! ELEMENT lname (# P(DATA)>
<! ELEMENT fname (# PCDATA) >
<! ELEMENT mname (# PCDATA)>
<! ELEMENT address (city, state, pincode )>
<! ELEMENT city (# PC DATA) >
<! ELEMENT state (# PCDATA)>
<! ELEMENT pincode (# PCDATA)>
<! ELEMENT mobile no. (Mob.)>
<! ELEMENT Mob. (# PCDATA)>
<! ELEMENT Recharge (Jan, Feb., March, April, May, June)
<! ELEMENT Recharge (# PCDATA)>
<! ELEMENT Jan (# PCDATA)>
<! ELEMENT Feb (# PCDATA)>
<! ELEMENT March (# PCDATA)>
<! ELEMENT April (# PCDATA)>
<! ELEMENT May (# PCDATA)>
<! ELEMENT June (# PCDATA)>
</information>
```

b) Differentiate b/w HTML and XML with the help of an example.

Teacher's Sign............................

Topic..Date...........................

Parameter	XML	HTML
Types of language	XML is a framework for specifying markup lang.	HTML is predefined markup language.
Language Type	Case sensitive	Case sensitive
Structural details	It is provided	It is not provided
Purpose	Transfer of data	Presentation of the Data
Codding errors	No coding errors are allowed	Small errors are ignored
Whitespace	You can use whitespaces in your code	You can't use white spaces in your code.
Nesting	should be done appropriately	Doesn't have any effect on the code.
Driven by	XML is content driven	HTML is format driven
End of Tags	The closing tag is essential in a well-formed XML document.	The closing tag is not always required.
Quotes	Quotes required around XML attribute values?	Quotes are not required for the values of attributes.
Object support	Objects have to be expressed by conventions. Mostly using attributes & elements.	Offer native object support.
Learning Curve	Very hard as you need to learn technologies like XPath, XML Schema, DOM etc.	HTML is a simple technology stack that is familiar to developers.

Teacher's Sign...........................

Topic..Date..............................

Example of XML

```
<?xml version = "1.0>
<address>
<name> Krishna Rungta </name>
<contact> 9896813050 </contact>
<email> Krishnaguru99@gmail.com </email>
<birthdate> 1985-09-07 </birthdate>
</address>
```

Example of HTML

```
<! DOCTYPE html>
<html>
<head>
<title> Page title </title> </head>
<body>
<h1> First heading </h1> <p> First paragraph </p>
</body>
</html>
```

Ques6: a) Under what conditions the use of entity beans is done? Describe the entity beans methods to update and destroy a database.

An entity bean represents a business object in a persistent storage mechanism. some examples of business objects are customers, orders and products. In the J2EE SDK, the persistent storage mechanism is a relational database. Typically, each entity bean has an underlying table in a relational database, and each instance of the bean corresponds to a row in that table.

You should probably use an entity bean under the following conditions:

Teacher's Sign..........................

Topic..Date..........................

- The bean represents a business entity, not a procedure. For example, Credit Card EJB would be an entity bean, but credit Card Verifier EJB would be a session bean.
- The bean's state must be persistent. If the bean instance terminates or if the J2EE server is shut down, the bean's state still exists in persistent storage (a database).

To update an Entity Bean Job, follow these steps:

1) select EJB Entity from the application services group on the job palette.
2) Drag and drop the EJB Entity icon onto the workspace.
3) Right-click the icon and select job details. The J2EE EJB Entity job dialog opens.
4) Click the agent specifications tab.
5) Complete the following required fields:
   - Name
   - Agent Name
   - Initial context factory
   - Provider URL
   - Bean name
   - Operation Type
   - Method Name
   - Finder Name.
6) (Optional) specify the following optional information:
   - User ID
   - Job class
7) Click Finder Parameters in the left pane. The finder Parameters page opens in the right pane.
8) Click New to define the parameters. The Edit Parameter dialog opens.

Teacher's Sign..........................

Topic..Date..............................

9) Complete the fields as required.
10) Click OK. The Edit parameter dialog closes and the parameter appears in the Parameters table.
11) Click OK. The New parameter dialog closes and the parameters appear in a list.
12) Click Modify Parameters in the left pane. The Modify parameters page opens in the right pane.
13) Repeat steps 7 through 10 to specify modify parameters.
14) Click OK. The Entity Bean job is defined.

Example : Update an Entity Bean
Suppose that you want to update the phone number for the acme company to 800-123-4567. The customer entity bean stores the customer id and phone number. The primary key for the customer is the customer id.
To update an entity bean, follow these steps:

1) Enter the following required information in the agent specifications tab :
- Name -- UPDATE
- Agent Name — APPAGENT
- Initial context factory — weblogic.jndi.WLInitialContext Factory
- Provider URL -- t3://localhost : 7001
- Bean name -- customer
- operation type -- UPDATE
- Method name -- changephone
- Finder name -- acme

2) Add the following value parameter in the Finder Parameters tab :
- Parameter Type -- String
- Parameter Value -- customerid

Teacher's Sign..........................

Topic...Date..............................

3) Add the following value parameter in the method Parameters tab:
- Parameter Type -- String
- Parameter Value -- 800-123-4567

4) Click OK. When the job runs, the acme company's phone number is changed.

b) Write the advantage of using an entity bean for database operations over directly using JDBC API.

The advantage of using an Entity Beans represents the data in a database. It isn't that Entity Beans replaces JDBC API.
There are two types of Entity Beans,
1. Container Managed
2. Bean Managed

1. Container - Managed Entity Bean:
Whenever the instance of the bean is created the container automatically retrieves the data from the database/ Persistence storage and assigns to the object variables in entity bean for the user to manipulate or use them. For this, the developer needs to map the fields in the database to the variables in deployment descriptor files.

2. Bean Managed Entity Bean: The developer has to specifically make the connection with the database, retrieve values, assign them to the objects in the ejbLoad () which will be called by the container when it instantiaties at entity bean object.

Teacher's Sign..........................

Topic...Date...............................

The ejbLoad and ejbStore are callback methods and can be only invoked by the container. Apart from this, when you use Entity beans you don't worry about database transaction handling, database connection pooling etc which are taken care of the EJB container.

Ques 7: With the help of an example, show implementation of a massage driven bean and explain the methods required to implement it.

A message driven bean is a type of enterprise bean, which is invoked by EJB container when it receives a message from queue or topic. Message driven bean is a stateless bean and is used to do task asynchronously. To demonstrate use of message driven bean, we will make use of EJB - persistence chapter and we need to do the following tasks -

- step 1 - Create table in database
- step 2 - Create entity class corresponding to table
- step 3 - Create data source and persistence unit
- step 4 - Create a stateless EJB having entity manager instance
- step 5 - Update stateless ejb. Add methods to add records and get records from database via entity manager.
- step 6 - Create a Queue named book Queue in JBoss default application directory.
- step 7 - A console based application client will send message to this queue.
- step 8 - Create a Message driven bean, which will use the stateless bean to persist the client data-

Teacher's Sign..........................

Topic.. Date..............................

- Step 9: EJB Container of jboss will call the above message driven bean and pass it the message that client will be sending to.

Example: To create the message driven bean, you need to declare @MessageDriven annotation and implement MessageListener interface. In eclipse ide, create EJB Project then create a class as given below:

```
packagecom.javatpoint;
importjavax.ejb.MessageDriven;
importjavax.jms.*;
@MessageDriven(mappedName = "mytopic")
public class MyListener implements MessageListener {
  @Override
public void onMessage(Message msg) {
  TextMessage m = (Text message) msg;
try {
  system.out.printen("message received: "+m.getText
  ());
} catch(Exception e) {system.out.printen(e); }
  }
   }
```

Export the ejb project and deploy the application. In glassfish server, click on applications → deploy → select mdb jar file by choose file → OK.

Ques 8: What are the advantages of using Java's multiple layer security implementation? Explain with the help of an example program.

Java EE security services are provided by the component container and can be implemented using declarative or programmatic techniques.

Teacher's Sign..........................

Topic...Date..............................

Java EE security services provide a robust and easily configured security mechanism for authenticating users and authorizing access to application functions and associated data at many different layers. Java EE security services are separate from the security mechanisms of the operating systems.

Application - Layer Security

In Java EE, component containers are responsible for providing application - layer security. Application - layer security provides security services for a specific application type tailored to the needs of the application. At the application layer, application firewalls can be employed to enhance application protection by protecting the communication stream and all associated application resources from attacks.

Java EE security is easy to implement and configure, and can offer fine - grained access control to application functions and data. However, as is inherent to security applied at the application layer, security properties are not transferable to applications running in other environments and only protect data while it is residing in the application environment.

The advantages of using application-layer security include the following:

- security is uniquely suited to the needs of the application.
- Security is fine - grained, with application specific settings.

Topic..Date..............................

Transport-layer security

Transport-layer security is provided by the transport mechanisms used to transmit information over the wire b/w clients and providers, thus transport-layer security relies on secure HTTP transport using secure sockets layer. Transport security is a point-to-point security mechanism that can be used for authentication, message integrity, and confidentiality.

Transport-layer security is performed in a series of phases, which are listed here:

- The client and server agree on an appropriate algorithm.
- A key is exchanged using public-key encryption and certificate-based authentication.
- A symmetric cipher is used during the information exchange.

The advantages of using transport-layer security include the following:

- Relatively simple, well understood, standard technology
- Applies to message body and attachments.

Message-Layer Security:

In message layer security, security information is contained within the SOAP message and/or SOAP message attachment; which allows security information to travel along with the message or attachment. For example, a portion of the message may be signed by a sender and encrypted for a particular receiver.

Teacher's Sign..........................

Topic..Date........................

The advantages of message-layer security include the following:

- security stays with the message over all hops and after the message arrives at its destination.
- security can be selectively applied to different portions of a message.
- Message security can be used with intermediaries over multiple hops.
- Message security is independent of the application environment or transport protocol.

Example:

```
public void service (HttpServlet Request request Http
 servelet Response response)
throws IOException, servlet Exception
{
// check to see if a session has already been
 created for this user.
// don't create a new session yet
HttpSession session = request. getSession (false);
String requestedPage = request. getParameter
 ( Constants. REQUEST);
if ( session! = null)
{
// retrieve authentication parameter from the
session
Boolean is Authenticated = ( Boolean)
session. getValue ( Constants. AUTHENTICATION);
// if the user is not authenticated
if ( ! is Authenticated . booleanValue ())
{
// process the unauthenticated request
```

Teacher's Sign..........................

Topic.......................................Date.......................

```
 unauthenticatedUser( response, requested page );
 }
}
else// the session does not exist
 {
 // therefore the user is not authenticated
 // process the unauthenticated request
 unauthenticated user ( response, requested page );
 }
  }
```

Ques 9: Discuss the security measures taken in SSL protocol.

secure sockets layer (SSL) is a networking protocol designed for securing connections b/w web clients and web servers over an insecure network, such as the internet. Netscape formally introduced the SSL protocol in 1995, making it the first widely used protocol for securing online transactions b/w consumers and businesses. It eventually came to be used to secure authentication and encryption for other applications at the network transport layer.

SSL suffered from numerous problems, and the Internet Engineering Task Force stopped recommending its use in 2015. It was replaced by the transport layer security (TLS) protocol. While SSL is still in use today, mostly in legacy systems, TLS has taken over its role in securing internet connections.

In addition to securing internet connections, SSL was also used to authenticate and encrypt

Teacher's Sign.......................

Topic...Date..................................

other applications at the network transport layer. SSL typically involved securing connections b/w a web browser (client) and a website (server). It facilitated safe transactions b/w consumers and businesses, helping create the foundation for e-commerce. Without SSL, data sent to and from a website could be intercepted by a threat actor.

SSL uses public key and private key encryption and other cryptographic functions to secure connections b/w devices communicating over a TCP/IP network. SSL can scramble clear text entered on a website using asymmetric cryptography and public key encryption. It is just one of the ways in which public keys infrastructure is used by modern businesses.

Teacher's Sign...........................