

# ASSIGNMENT-3

CSA4307

INTERNET

PROGRAMMING.

- JAYA HARINI. J.

192371034.

BTech CSB105.



JDBC program to Retrieve and display All Records from the Employee Table using Type 2 Driver.

import java.sql.

Public class EmployeeQuery

{

Public static void main (String[] args)

{

connection conn = null;

Statement stmt = null;

ResultSet rs = null;

try

{

class.forName ("oracle.jdbc.driver.OracleDriver");

String url = "jdbc:oci:";

String user = "scott";

String password = "tiger";

conn = DriverManager.getConnection (url, user, password);

System.out.println ("Employee ID \t Name");

while (rs.next()) {

int id = rs.getInt ("EmployeeID");

String name = rs.getString ("Name");

System.out.println (id + "\t\t" + name);

}

} catch (ClassNotFoundException e)

{

System.out.println ("Oracle JDBC driver not found");

e.printStackTrace();

} catch (SQLException e)



```

{
    system.out.println("Database error");
    e.printStackTrace();
} finally
{
    try {
        if (rs != null) rs.close();
        if (stmt != null) stmt.close();
        if (conn != null) conn.close();
    } catch (SQLException se)
    {
        se.printStackTrace();
    }
}
}

```

Assume.

Employee ID	Name
101	Alice
102	Bob
103	Charlie



# Managing user preferences with cookies in Web Application.

## Types of cookies & their use:

### 1. Session cookies

i) Definition: Stored in memory and deleted when the browser is closed.

ii) use case: Ideal for temporary data.

### 2. Persistent cookies.

i) Definition: Stored on the user's device with an expiration date

ii) use case: useful for remembering user preferences, login information, and theme choices.

## How cookies improve user Experience.

i) personalization: Remember language, theme, or layout settings.

ii) Convenience: keep users logged in.

iii) Efficiency: Store preferences to reduce loading.

## Javascript code.

### COOKIE.

function setcookie (name, value, days)

```
{
  let expires = "";
  if (days) {
    let date = new Date();
    date.setTime(date.getTime() + (days * 24 * 60 * 60 * 1000));
    expires = "; expires = " + date.toUTCString();
  }
}
```



```
document.cookie = name + "=" + encodeURIComponent (value) + expires + "; path = /";
```

```
}
```

Get a cookie.

```
function getcookie (name)
{
  let decodedcookie = decodeURIComponent (document.cookie);
  let cookies = decodedcookie.split(';');
  name = name + '=';
  for (let cookie of cookies)
  {
    while (cookie.charAt(0) === ' ') cookie = cookie
      .substring(1);
    if (cookie.indexOf (name) === 0)
    {
      return cookie.substring (name.length, cookie.
        length);
    }
  }
  return "";
}
```

Example usage.

```
Setcookie (" theme", "dark", 1);
let usertheme = getcookie ("theme");
console.log ("Selected Theme : " + userTheme);
```



# Shopping cart Management using Java Servlets and JSP.

## Design Overview.

- i) cart structure: use a Java class (e.g. `CartItem`) to represent individual items.
- ii) cart storage: store a `List < CartItem >` or `Map < Product ID, CartItem >`.
- iii) Servlets: Handle add / remove / update actions;

## CartItem.java.

```
Public class CartItem {  
    Private String productId;  
    Private String name;  
    Private int quantity;  
    Private double price;  
    Public CartItem (String productId, String name, int quantity)  
    {  
        this.productId = productId;  
        this.name = name;  
        this.quantity = quantity;  
    }  
}
```

## 3. CartServlet.java

```
Public class CartServlet {  
    Protected void doPost (HttpServletRequest request,  
        HttpServletResponse response) throws ServletException  
        Exception)
```



```
if (cart == null)
    cart = new ArrayList();
```

```
}
```

```
String action = request.getParameter("action");
String productId = request.getParameter("ProductId");
String name = request.getParameter("name");
int quantity = Integer.parseInt(request.getParameter("quantity"));
```

```
if ("add".equals(action))
```

```
{
```

```
    boolean found = false;
```

```
    for (CartItem item : cart)
```

```
    {
```

```
        if (item.getProductId().equals(productId))
```

```
            found = true;
```

```
            break;
```

```
    }
```

```
    if (!found)
```

```
    { cart.add(new CartItem(productId, name));
```

```
    }
```

```
else
```

```
{
```

```
    session.setAttribute("cart", cart);
```

```
    response.sendRedirect("cart.jsp");
```

```
}
```

```
}
```