FREE Live Classes

Java

Theory

Coding

RedSysTech
-Love to Learn...

Core Java

by Karthik Ponnusamy

Course Contents:-

- Core Java Basic Concepts
- OOP Concepts
- Fundamentals of Java Programming
- Java Control Flow Statements
- Deep dive into Collection Framework
- Deep dive into Multithreading
- Java 1.8 Features

Batch Starts from Jan - 18 - 2022

Mon to Thurs - 10 PM to 11 PM CST /
9:30 AM to 10:30 AM IST

Day 06: Agenda

Sample Java Web Application using Servlets and JSP

Practical Hand-on project by creating a simple java web application using JSP and Servlet

Project Overview

In this project, we will create a simple web application that allows users to view and add books to a collection. The application will have two main pages: a homepage that displays a list of books, and a form page that allows users to add new books to the collection.

Prerequisites

To follow along with this project, you will need to have the following software installed on your computer:

1. Java Development Kit (JDK) version 8 or higher
2. Apache Tomcat server version 8 or higher
3. Eclipse IDE for Java EE Developers

Project Steps

Create a new Dynamic Web Project in Eclipse:

Open Eclipse and select "File" > "New" > "Dynamic Web Project".

Enter a project name (e.g. "BookCollection") and click "Next".

Select "Generate web.xml deployment descriptor" and click "Finish".

Create a database table:

Open the MySQL command line client and create a new database called “bookdb": **CREATE DATABASE bookdb;**

Create a table called "books" with columns for "id", "title", and “author":

CREATE TABLE books (id INT PRIMARY KEY AUTO_INCREMENT, title VARCHAR(50), author VARCHAR(50));

Add the MySQL Connector/J JAR file to the project:

Download the latest version of the MySQL Connector/J JAR file from the official MySQL website.

In Eclipse, right-click on the project and select "Build Path" > "Configure Build Path".

Click on the "Libraries" tab and then click "Add External JARs".

Select the MySQL Connector/J JAR file and click "Open".

Create a Java Servlet:

Right-click on the project and select "New" > "Servlet".

Enter a name for the servlet (e.g. "BookServlet") and click "Next".

Enter a package name (e.g. "com.example.bookcollection") and click "Next".

In the "Web content folder" section, select "Use project folder" and click "Finish".

In the doGet() method of the servlet, retrieve the list of books from the database and store it in a request attribute:

```
// Get a connection to the database
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/bookdb", "root",
"password");
// Create a statement
Statement stmt = conn.createStatement();
// Execute a SELECT query
ResultSet rs = stmt.executeQuery("SELECT * FROM books");
// Create a list of books
List<Book> books = new ArrayList<>();
while (rs.next())
{
   int id = rs.getInt("id");
   String title = rs.getString("title");
   String author = rs.getString("author");
   books.add(new Book(id, title, author));
}
 // Store the list of books in a request attribute
   request.setAttribute("books", books);
// Forward the request to the JSP page
   RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");
   dispatcher.forward(request, response);
```

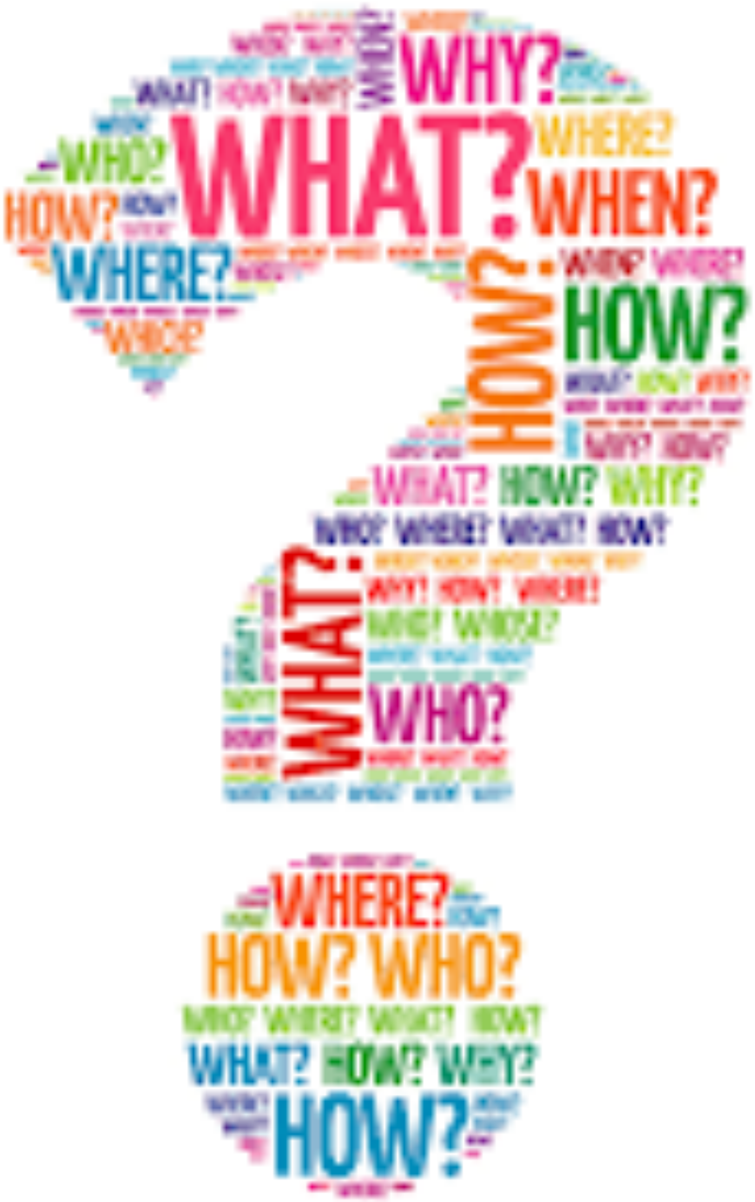
Create a JSP page:

Right-click on the project and select "New" > "JSP File".

Enter a name for the JSP page (e.g. "index.jsp") and click "Finish".

In the JSP page, display the list of books using a <c:forEach> loop:

```
<c:forEach items
```


WHY?
WHAT?
WHERE?
WHEN?
WHO?
HOW?
WHERE?
HOW?
HOW?
WHAT? HOW? WHY?
WHO? WHERE? WHAT? HOW?
WHAT?
WHO?
WHERE?
HOW? WHO?
WHAT? HOW? WHY?
HOW?