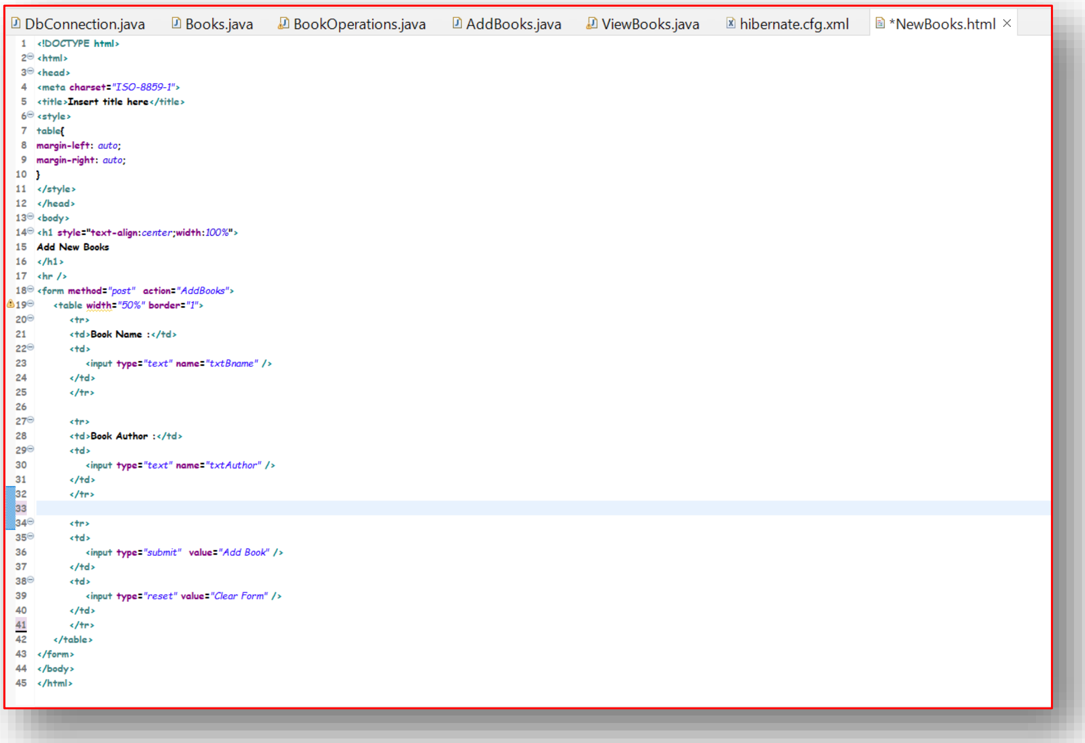
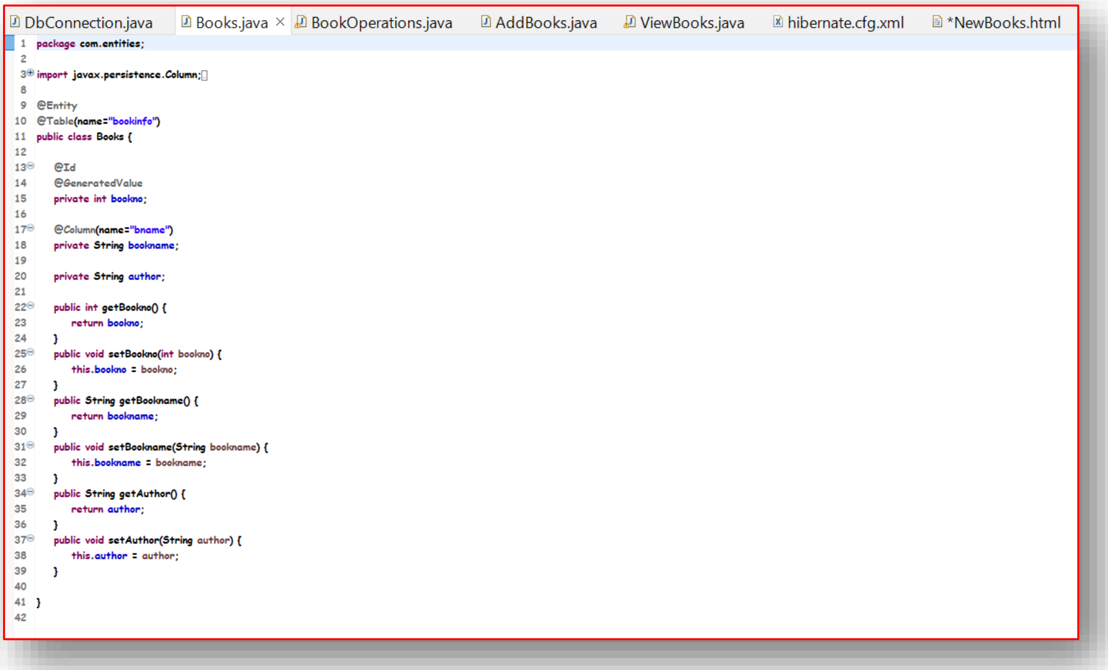
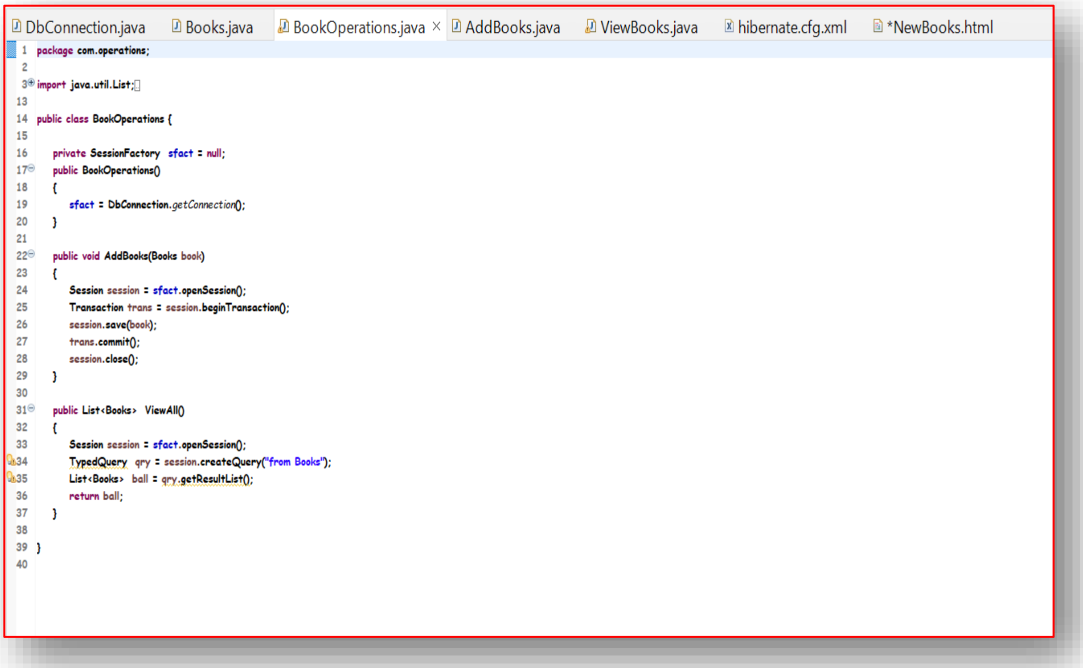
Source Code



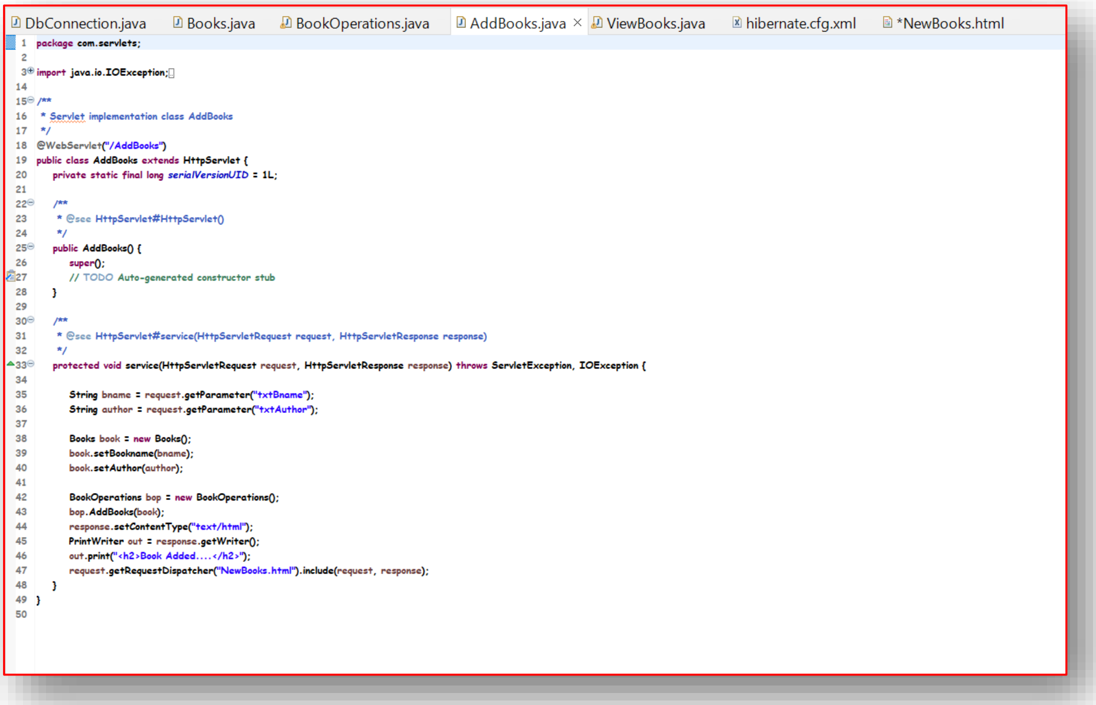
```html
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<style>
table{
margin-left: auto;
margin-right: auto;
}
</style>
</head>
<body>
<h1 style="text-align:center;width:100%">
Add New Books
</h1>
<hr />
<form method="post"  action="AddBooks">
    <table width="50%" border="1">
        <tr>
        <td>Book Name :</td>
        <td>
            <input type="text" name="txtBname" />
        </td>
        </tr>

        <tr>
        <td>Book Author :</td>
        <td>
            <input type="text" name="txtAuthor" />
        </td>
        </tr>

        <tr>
        <td>
            <input type="submit"  value="Add Book" />
        </td>
        <td>
            <input type="reset" value="Clear Form" />
        </td>
        </tr>
    </table>
</form>
</body>
</html>
```



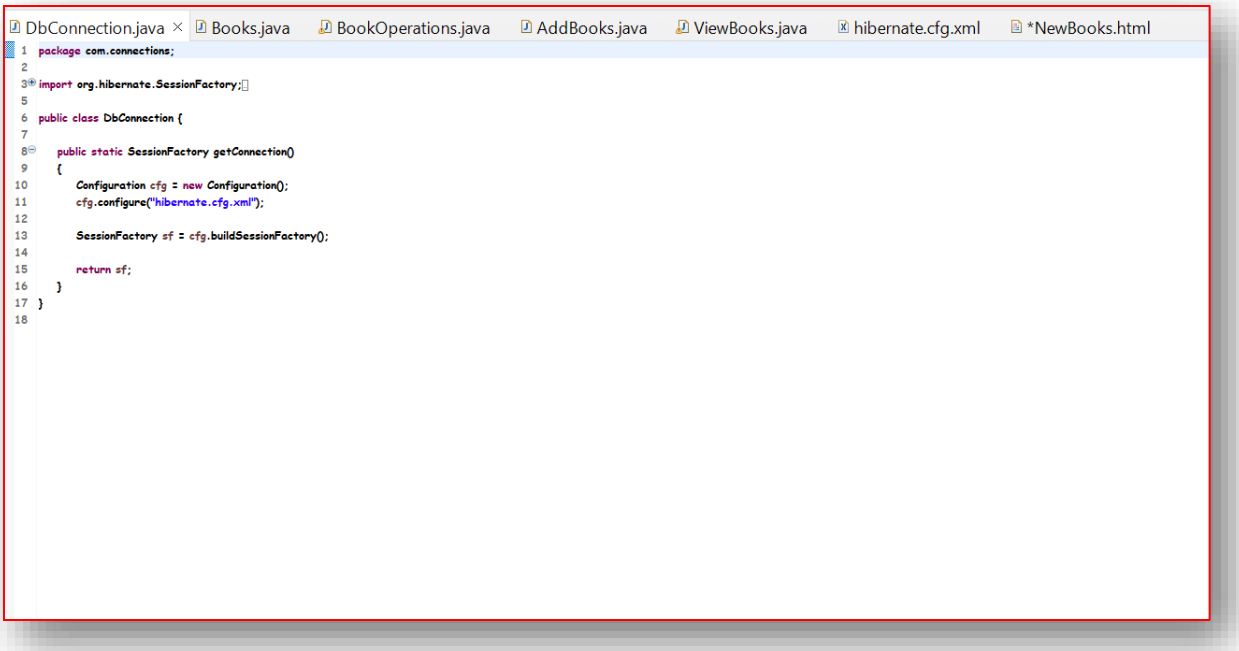
```java
package com.entities;

import javax.persistence.Column;

@Entity
@Table(name="bookinfo")
public class Books {

    @Id
    @GeneratedValue
    private int bookno;

    @Column(name="bname")
    private String bookname;

    private String author;

    public int getBookno() {
        return bookno;
    }
    public void setBookno(int bookno) {
        this.bookno = bookno;
    }
    public String getBookname() {
        return bookname;
    }
    public void setBookname(String bookname) {
        this.bookname = bookname;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }

}

```



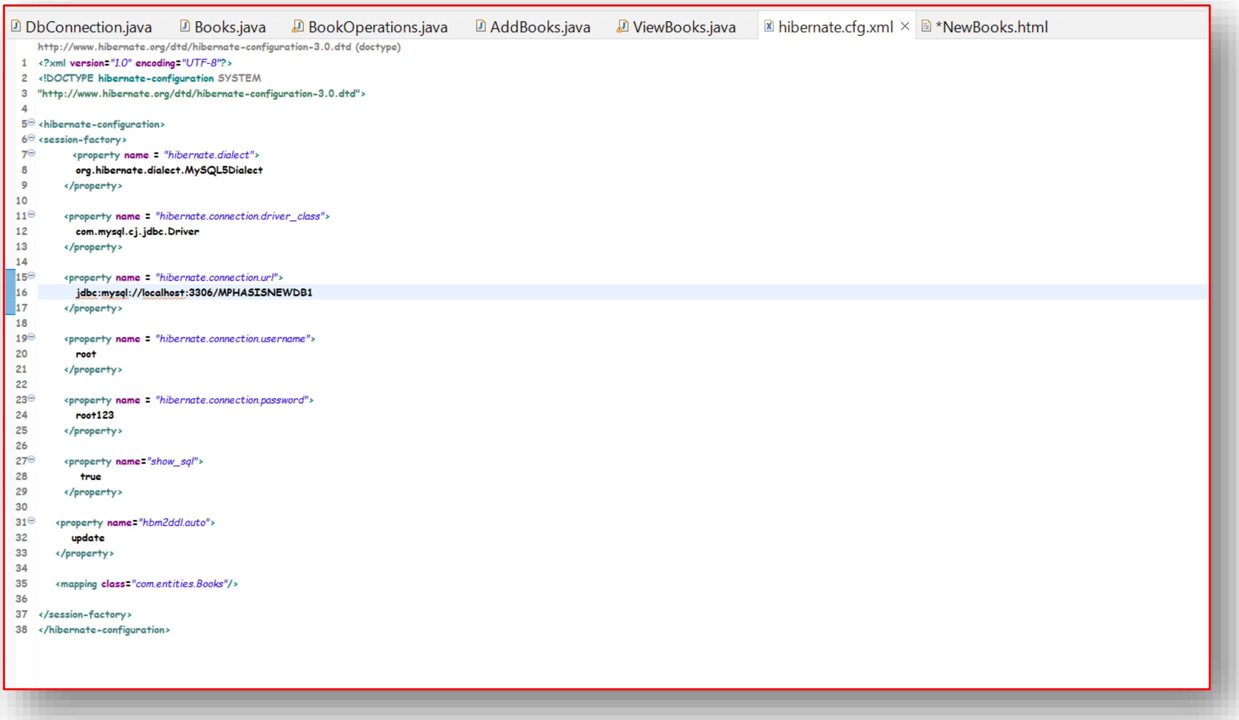
```java
package com.operations;

import java.util.List;

public class BookOperations {

    private SessionFactory  sfact = null;
    public BookOperations()
    {
        sfact = DbConnection.getConnection();
    }

    public void AddBooks(Books book)
    {
        Session session = sfact.openSession();
        Transaction trans = session.beginTransaction();
        session.save(book);
        trans.commit();
        session.close();
    }

    public List<Books>  ViewAll()
    {
        Session session = sfact.openSession();
        TypedQuery  qry = session.createQuery("from Books");
        List<Books>  ball = qry.getResultList();
        return ball;
    }

}
```



```java
package com.servlets;

import java.io.IOException;

/**
 * Servlet implementation class AddBooks
 */
@WebServlet("/AddBooks")
public class AddBooks extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public AddBooks() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String bname = request.getParameter("txtBname");
        String author = request.getParameter("txtAuthor");

        Books book = new Books();
        book.setBookname(bname);
        book.setAuthor(author);

        BookOperations bop = new BookOperations();
        bop.AddBooks(book);
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.print("<h2>Book Added....</h2>");
        request.getRequestDispatcher("NewBooks.html").include(request, response);
    }
}
```



```java
package com.connections;

import org.hibernate.SessionFactory;

public class DbConnection {

    public static SessionFactory getConnection()
    {
        Configuration cfg = new Configuration();
        cfg.configure("hibernate.cfg.xml");

        SessionFactory sf = cfg.buildSessionFactory();

        return sf;
    }
}

```



```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration SYSTEM
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
<session-factory>
        <property name = "hibernate.dialect">
         org.hibernate.dialect.MySQL5Dialect
      </property>

      <property name = "hibernate.connection.driver_class">
         com.mysql.cj.jdbc.Driver
      </property>

      <property name = "hibernate.connection.url">
         jdbc:mysql://localhost:3306/MPHASISNEWDB1
      </property>

      <property name = "hibernate.connection.username">
         root
      </property>

      <property name = "hibernate.connection.password">
         root123
      </property>

      <property name="show_sql">
          true
      </property>

    <property name="hbm2ddl.auto">
        update
    </property>

    <mapping class="com.entities.Books"/>

</session-factory>
</hibernate-configuration>
```