























JAVA SERVER FACES  
LESSONS – MATERIALS.1









## Линкови

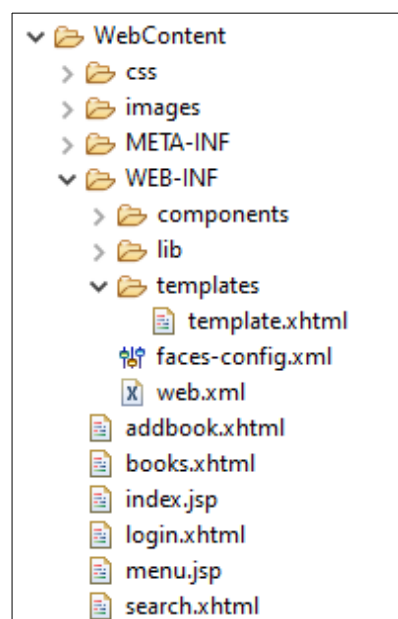
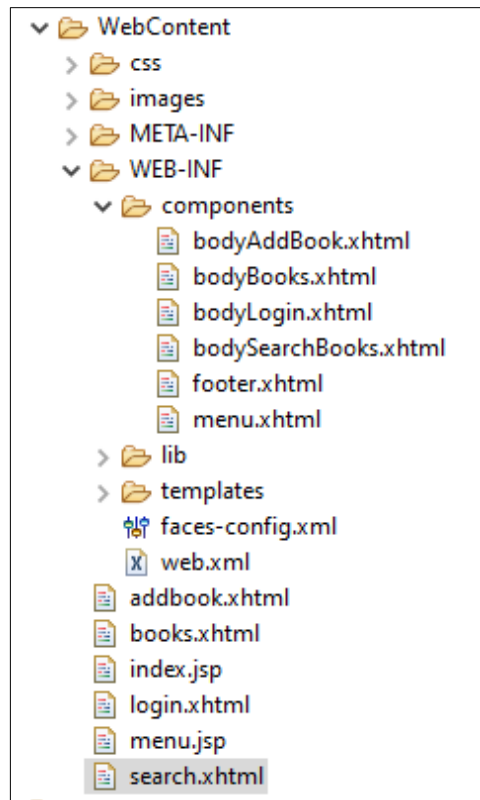
[addbook.xhtml](#)  
[books.xhtml](#)  
[css/bluebliss.css](#)  
[images/footer.gif](#)  
[images/header.gif](#)  
[images/tab.gif](#)  
[images/tab2.gif](#)  
[index.jsp](#)  
[login.xhtml](#)  
[menu.jsp](#)  
[META-INF/MANIFEST.MF](#)  
[search.xhtml](#)  
[WEB-INF\components\bodyAddBook.xhtml](#)  
[WEB-INF\components\bodyBooks.xhtml](#)  
[WEB-INF\components\bodyLogin.xhtml](#)  
[WEB-INF\components\bodySearchBooks.xhtml](#)  
[WEB-INF\components\footer.xhtml](#)  
[WEB-INF\components\menu.xhtml](#)  
[WEB-INF\faces-config.xml](#)  
[WEB-INF\lib\javax.faces-2.2.4.jar](#)  
[WEB-INF\lib\jstl-api-1.2.jar](#)  
[WEB-INF\lib\jstl-impl-1.2.jar](#)  
[WEB-INF\lib\mysql-connector-java-8.0.22.jar](#)  
[WEB-INF\lib\omnifaces-1.7.jar](#)  
[WEB-INF\templates\template.xhtml](#)  
[WEB-INF\web.xml](#)

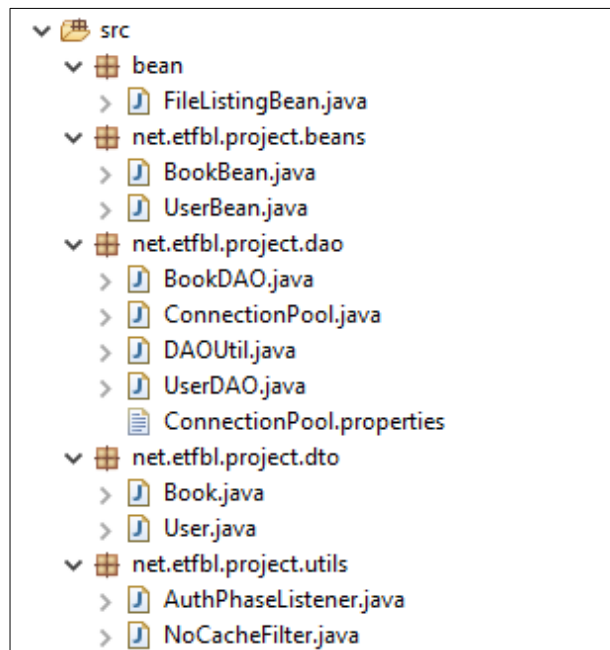
Пребројано је 26 ставки.

- ▼  src
  - >  bean
  - >  net.etfbl.project.beans
  - >  net.etfbl.project.dao
  - >  net.etfbl.project.dto
  - >  net.etfbl.project.utils

- ▼  WebContent
  - >  css
  - >  images
  - >  META-INF
  - ▼  WEB-INF
    - >  components
    - >  lib
    - >  templates
    -  faces-config.xml
    -  web.xml
    -  addbook.xhtml
    -  books.xhtml
    -  index.jsp
    -  login.xhtml
    -  menu.jsp
    -  search.xhtml

- ▼  WEB-INF
  - >  components
  - ▼  lib
    -  javax.faces-2.2.4.jar
    -  jstl-api-1.2.jar
    -  jstl-impl-1.2.jar
    -  mysql-connector-java-8.0.22.jar
    -  omnifaces-1.7.jar





## WEB.XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation='
3   <display-name>046_IP_Predavanja ETF_IP_JSF_LAB</display-name>
4 <servlet>
5   <servlet-name>Faces Servlet</servlet-name>
6   <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
7   <load-on-startup>1</load-on-startup>
8 </servlet>
9 <servlet-mapping>
10  <servlet-name>Faces Servlet</servlet-name>
11  <url-pattern>/faces/*</url-pattern>
12 </servlet-mapping>
13 <context-param>
14   <description>State saving method: 'client' or 'server' (=default). See JSF Specification 2.5.2</description>
15   <param-name>javax.faces.STATE_SAVING_METHOD</param-name>
16   <param-value>client</param-value>
17 </context-param>
18 <context-param>
19   <param-name>javax.servlet.jsp.jstl.fmt.localizationContext</param-name>
20   <param-value>resources.application</param-value>
21 </context-param>
22 <listener>
23   <listener-class>com.sun.faces.config.ConfigureListener</listener-class>
24 </listener>
25 <filter>
26   <filter-name>NoCacheFilter</filter-name>
27   <filter-class>net.etfbl.project.utils.NoCacheFilter</filter-class>
28 </filter>
29 <filter-mapping>
30   <filter-name>NoCacheFilter</filter-name>
31   <servlet-name>Faces Servlet</servlet-name>
32 </filter-mapping>
33 </web-app>

```

## FACES-CONFIG.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<faces-config
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-facesconfig_2_2.xsd" version="2.2">
  <managed-bean>
    <managed-bean-name>bookBean</managed-bean-name>
    <managed-bean-class>net.etfbl.project.beans.BookBean</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean>
  <managed-bean>
    <managed-bean-name>userBean</managed-bean-name>
    <managed-bean-class>net.etfbl.project.beans.UserBean</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean>
  <lifecycle>
    <phase-listener>net.etfbl.project.utils.AuthPhaseListener</phase-listener>
  </lifecycle>
</faces-config>
```

## ConnectionPool.properties

```
1 jdbcURL=jdbc:mysql://127.0.0.1:3306/etf?useUnicode=true&characterEncoding=UTF-8
2 username=root
3 password=root
4 driver=com.mysql.jdbc.Driver
5 preconnectCount=2
6 maxIdleConnections=25
7 maxConnections=25
```

## DAOUtil.java

```
1 package net.etfbl.project.dao;
2
3 import java.sql.*;
4
5 public final class DAOUtil {
6
7     public static PreparedStatement prepareStatement(Connection connection,
8         String sql, boolean returnGeneratedKeys, Object... values)
9         throws SQLException {
10         PreparedStatement preparedStatement = connection.prepareStatement(sql,
11             returnGeneratedKeys ? Statement.RETURN_GENERATED_KEYS
12             : Statement.NO_GENERATED_KEYS);
13         setValues(preparedStatement, values);
14         return preparedStatement;
15     }
16
17     public static void setValues(PreparedStatement preparedStatement,
18         Object... values) throws SQLException {
19         for (int i = 0; i < values.length; i++) {
20             preparedStatement.setObject(i + 1, values[i]);
21         }
22     }
23 }
24
```

## NoCacheFilter.java

```

1 package net.etfbl.project.utils;
2
3 import java.io.IOException;
4
15 public class NoCacheFilter implements Filter {
16
17     @Override
18     public void destroy() {
19         // TODO Auto-generated method stub
20     }
21
22
23     @Override
24     public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
25         HttpServletRequest httpReq = (HttpServletRequest) request;
26         HttpServletResponse httpRes = (HttpServletResponse) response;
27
28         if (!httpReq.getRequestURI().startsWith(httpReq.getContextPath() + ResourceHandler.RESOURCE_IDENTIFIER)) {
29             httpRes.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
30             httpRes.setHeader("Pragma", "no-cache");
31             httpRes.setDateHeader("Expires", 0);
32         }
33
34         chain.doFilter(request, response);
35     }
36
37     @Override
38     public void init(FilterConfig arg0) throws ServletException {
39         // TODO Auto-generated method stub
40     }
41
42 }
43
44

```

## AuthPhaseListener.java

```

1 package net.etfbl.project.utils;
2
3 import java.util.Map;
4
12 public class AuthPhaseListener implements PhaseListener {
13     private static final long serialVersionUID = 1L;
14
15
16     @SuppressWarnings("rawtypes")
17     public void afterPhase(PhaseEvent event) {
18         FacesContext facesContext = event.getFacesContext();
19         ExternalContext context = facesContext.getExternalContext();
20         context.getSession(true);
21         Map sessMap = context.getSessionMap();
22
23         UserBean userBean = (UserBean) sessMap.get("userBean");
24
25         boolean onLoginPage = (-1 != facesContext.getViewRoot().getViewId().lastIndexOf("login")) ? true : false;
26         if (!onLoginPage && ((userBean == null) || !(userBean.isLoggedin()))) {
27             event.getFacesContext().getApplication().getNavigationHandler().handleNavigation(event.getFacesContext(), null, "login");
28         }
29     }
30
31     public void beforePhase(PhaseEvent event) {
32     }
33
34
35     public PhaseId getPhaseId() {
36         return PhaseId.RESTORE_VIEW;
37     }
38
39 }
40

```

Timer  
 TimerTask  
 Concurrent  
 Concurrent Basic  
 Java FX Swing AWT JFC  
 Generics JCF

## User.java

```
1 package net.etfbl.project.dto;
2
3 public class User {
4     private int id;
5     private String username;
6     private String password;
7     private int userGroup;
8     public User() {
9     }
10    public User(int id, String username, String password, int userGroup) {
11        super();
12        this.id = id;
13        this.username = username;
14        this.password = password;
15        this.userGroup = userGroup;
16    }
17    public int getId() {
18        return id;
19    }
20    public void setId(int id) {
21        this.id = id;
22    }
23    public String getUsername() {
24        return username;
25    }
26    public void setUsername(String username) {
27        this.username = username;
28    }
29    public String getPassword() {
30        return password;
31    }
32    public void setPassword(String password) {
33        this.password = password;
34    }
35    public int getUserGroup() {
36        return userGroup;
37    }
38    public void setUserGroup(int userGroup) {
39        this.userGroup = userGroup;
40    }
41 }
42
```

## Book.java

```
1 package net.etfbl.project.dto;
2
3 import java.io.Serializable;
4
5 public class Book implements Serializable{
6     /**
7      *
8      */
9     private static final long serialVersionUID = 1L;
10    private int id;
11    private String author;
12    private String title;
13    private String publisher;
14    private int year;
15    private String isbn;
16    public Book() {
17    }
18    public Book(String author, String title,
19                String publisher, int year, String isbn) {
20        super();
21        this.author = author;
22        this.title = title;
23        this.publisher = publisher;
24        this.year = year;
25        this.isbn = isbn;
26    }
27    public Book(int id, String author, String title, String publisher,
28                int year, String isbn) {
29        super();
30        this.id = id;
31        this.author = author;
32        this.title = title;
33        this.publisher = publisher;
34        this.year = year;
35        this.isbn = isbn;
36    }
```



```
36     }
37     public int getId() {
38         return id;
39     }
40     public void setId(int id) {
41         this.id = id;
42     }
43     public String getAuthor() {
44         return author;
45     }
46     public void setAuthor(String author) {
47         this.author = author;
48     }
49     public String getTitle() {
50         return title;
51     }
52     public void setTitle(String title) {
53         this.title = title;
54     }
55     public String getPublisher() {
56         return publisher;
57     }
58     public void setPublisher(String publisher) {
59         this.publisher = publisher;
60     }
61     public int getYear() {
62         return year;
63     }
64     public void setYear(int year) {
65         this.year = year;
66     }
67     public String getIsbn() {
68         return isbn;
69     }
70     public void setIsbn(String isbn) {
71         this.isbn = isbn;
72     }
73 }
74
```

## ConnectionPool.java

```
1 package net.etfbl.project.dao;
2
3 import java.sql.*;
4
5
6 public class ConnectionPool {
7
8     public static ConnectionPool getConnectionPool() {
9         return connectionPool;
10    }
11
12    private static ConnectionPool connectionPool;
13
14    static {
15        ResourceBundle bundle =
16            PropertyResourceBundle.getBundle("net.etfbl.project.dao.ConnectionPool");
17        String jdbcURL = bundle.getString("jdbcURL");
18        String username = bundle.getString("username");
19        String password = bundle.getString("password");
20        String driver = bundle.getString("driver");
21        int preconnectCount = 0;
22        int maxIdleConnections = 10;
23        int maxConnections = 10;
24        try {
25            Class.forName(driver);
26            preconnectCount = Integer.parseInt(
27                bundle.getString("preconnectCount"));
28            maxIdleConnections = Integer.parseInt(
29                bundle.getString("maxIdleConnections"));
30            maxConnections = Integer.parseInt(
31                bundle.getString("maxConnections"));
32        } catch (Exception ex) {
33            ex.printStackTrace();
34        }
35        try {
36            connectionPool = new ConnectionPool(
37                jdbcURL, username, password,
38                preconnectCount, maxIdleConnections,
39                maxConnections);
40        } catch (Exception ex) {
41            ex.printStackTrace();
42        }
43    }
44 }
```

```

45 protected ConnectionPool(String aJdbcURL, String aUsername,
46     String aPassword, int aPreconnectCount,
47     int aMaxIdleConnections,
48     int aMaxConnections)
49     throws ClassNotFoundException, SQLException {
50
51     freeConnections = new Vector<Connection>();
52     usedConnections = new Vector<Connection>();
53     jdbcURL = aJdbcURL;
54     username = aUsername;
55     password = aPassword;
56     preconnectCount = aPreconnectCount;
57     maxIdleConnections = aMaxIdleConnections;
58     maxConnections = aMaxConnections;
59
60     for (int i = 0; i < preconnectCount; i++) {
61         Connection conn = DriverManager.getConnection(
62             jdbcURL, username, password);
63         conn.setAutoCommit(true);
64         freeConnections.addElement(conn);
65     }
66     connectCount = preconnectCount;
67 }

```

```

69 public synchronized Connection checkOut()
70     throws SQLException {
71
72     Connection conn = null;
73     if (freeConnections.size() > 0) {
74         conn = (Connection)freeConnections.elementAt(0);
75         freeConnections.removeElementAt(0);
76         usedConnections.addElement(conn);
77     } else {
78         if (connectCount < maxConnections) {
79             conn = DriverManager.getConnection(
80                 jdbcURL, username, password);
81             usedConnections.addElement(conn);
82             connectCount++;
83         } else {
84             try {
85                 wait();
86                 conn = (Connection)freeConnections.elementAt(0);
87                 freeConnections.removeElementAt(0);
88                 usedConnections.addElement(conn);
89             } catch (InterruptedException ex) {
90                 ex.printStackTrace();
91             }
92         }
93     }
94     return conn;
95 }
96
97 public synchronized void checkIn(Connection aConn) {
98     if (aConn == null)
99         return;
100     if (usedConnections.removeElement(aConn)) {
101         freeConnections.addElement(aConn);
102         while (freeConnections.size() > maxIdleConnections) {
103             int lastOne = freeConnections.size() - 1;
104             Connection conn = (Connection)
105                 freeConnections.elementAt(lastOne);
106             try { conn.close(); } catch (SQLException ex) { }
107             freeConnections.removeElementAt(lastOne);
108         }
109         notify();
110     }
111 }
112

```

```
112
113     private String jdbcURL;
114     private String username;
115     private String password;
116     private int preconnectCount;
117     private int connectCount;
118     private int maxIdleConnections;
119     private int maxConnections;
120     private Vector<Connection> usedConnections;
121     private Vector<Connection> freeConnections;
122
123 }
```

## BookDAO.java

```

1 package net.etfbl.project.dao;
2
3 import java.sql.Connection;
4
5 public class BookDAO {
6     private static ConnectionPool connectionPool = ConnectionPool.getConnectionPool();
7     private static final String SQL_SELECT_ALL = "SELECT * FROM book";
8     private static final String SQL_DELETE_BY_ID = "DELETE FROM book where id = ?";
9     private static final String SQL_INSERT = "INSERT INTO book (author, title, publisher, year, isbn) VALUES (?, ?, ?, ?, ?)";
10    private static final String SQL_SELECT_BY_TITLE = "SELECT * FROM book WHERE title LIKE ?";
11
12    public static ArrayList<Book> selectAll(){
13        ArrayList<Book> retVal = new ArrayList<Book>();
14        Connection connection = null;
15        ResultSet rs = null;
16        Object values[] = {};
17        try {
18            connection = connectionPool.checkOut();
19            PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
20                SQL_SELECT_ALL, false, values);
21            rs = pstmt.executeQuery();
22            while (rs.next()){
23                retVal.add(new Book(rs.getInt("id"), rs.getString("author"), rs.getString("title"), rs.getString("publisher"), r
24            }
25            pstmt.close();
26        } catch (SQLException exp) {
27            exp.printStackTrace();
28        } finally {
29            connectionPool.checkIn(connection);
30        }
31        return retVal;
32    }
33 }

```

```
rs.getString("title"), rs.getString("publisher"), rs.getInt("year"),rs.getString("isbn"))));
```

```

40 public static boolean insert(Book book) {
41     boolean retVal = false;
42     Connection connection = null;
43     ResultSet generatedKeys = null;
44     Object values[] = { book.getAuthor(), book.getTitle(), book.getPublisher(), book.getYear(), book.getIsbn()};
45     try {
46         connection = connectionPool.checkOut();
47         PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_INSERT, true,
48             values);
49         int affectedRows = pstmt.executeUpdate();
50         if (affectedRows == 0)
51             retVal = false;
52         else
53             retVal = true;
54         generatedKeys = pstmt.getGeneratedKeys();
55         if (generatedKeys.next())
56             book.setId(generatedKeys.getInt(1));
57         pstmt.close();
58     } catch (SQLException e) {
59         retVal = false;
60     } finally {
61         connectionPool.checkIn(connection);
62     }
63     return retVal;
64 }
65

```

```

66 public static boolean delete(Book book) {
67     boolean retVal = false;
68     Connection connection = null;
69     Object values[] = { book.getId()};
70     try {
71         connection = connectionPool.checkOut();
72         PreparedStatement pstmt = DAOUtil.prepareStatement(connection, SQL_DELETE_BY_ID, true,
73             values);
74         int affectedRows = pstmt.executeUpdate();
75         if (affectedRows == 0)
76             retVal = false;
77         else
78             retVal = true;
79         pstmt.close();
80     } catch (SQLException e) {
81         retVal = false;
82     } finally {
83         connectionPool.checkIn(connection);
84     }
85     return retVal;
86 }
87

```

```

88 public static ArrayList<Book> selectByTitleSearch(String titleSearch){
89     ArrayList<Book> retVal = new ArrayList<Book>();
90     Connection connection = null;
91     ResultSet rs = null;
92     Object values[] = {"%"+titleSearch+"%"};
93     try {
94         connection = connectionPool.checkOut();
95         PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
96             SQL_SELECT_BY_TITLE, false, values);
97         rs = pstmt.executeQuery();
98         while (rs.next()){
99             retVal.add(new Book(rs.getInt("id"), rs.getString("author"), rs.getString("title"),
100             }
101             pstmt.close();
102         } catch (SQLException exp) {
103             exp.printStackTrace();
104         } finally {
105             connectionPool.checkIn(connection);
106         }
107         return retVal;
108     }
109 }

```

## UserDAO.jsp

```
1 package net.etfbl.project.dao;
2
3 import java.sql.Connection;
4
5
6
7
8
9
10
11 public class UserDAO {
12     private static ConnectionPool connectionPool = ConnectionPool.getConnectionPool();
13     private static final String SQL_SELECT_BY_USERNAME_AND_PASSWORD = "SELECT * FROM user WHERE username=? AND password=?";
14
15     public static User selectByUsernameAndPassword(String username, String password){
16         User user = null;
17         Connection connection = null;
18         ResultSet rs = null;
19         Object values[] = {username, password};
20         try {
21             connection = connectionPool.checkOut();
22             PreparedStatement pstmt = DAOUtil.prepareStatement(connection,
23                 SQL_SELECT_BY_USERNAME_AND_PASSWORD, false, values);
24             rs = pstmt.executeQuery();
25             if (rs.next()){
26                 user = new User(rs.getInt("id"), rs.getString("username"), rs.getString("password"), rs.getInt("user_group"));
27             }
28             pstmt.close();
29         } catch (SQLException exp) {
30             exp.printStackTrace();
31         } finally {
32             connectionPool.checkIn(connection);
33         }
34         return user;
35     }
36 }
37
38
```

## UserBean.jsp

```
1 package net.etfbl.project.beans;
2
3 import net.etfbl.project.dao.UserDAO;
4
5
6 public class UserBean {
7     private User user = new User();
8     private boolean loggedIn = false;
9
10    public String login(){
11        if ((user = UserDAO.selectByUsernameAndPassword(user.getUsername(), user.getPassword()))!=null){
12            loggedIn = true;
13            return "books?faces-redirect=true";
14        }
15        user = new User();
16        loggedIn = false;
17        return "login?faces-redirect=true";
18    }
19
20    public User getUser() {
21        return user;
22    }
23
24    public boolean isLoggedIn() {
25        return loggedIn;
26    }
27
28    public String logout(){
29        loggedIn = false;
30        return "login?faces-redirect=true";
31    }
32 }
33
```

## BookBean.jsp

```
1 package net.etfbl.project.beans;
2
3+ import net.etfbl.project.dao.UserDAO;
4
5
6 public class UserBean {
7     private User user = new User();
8     private boolean loggedIn = false;
9
10    public String login(){
11        if ((user = UserDAO.selectByUsernameAndPassword(user.getUsername(), user.getPassword()))!=null){
12            loggedIn = true;
13            return "books?faces-redirect=true";
14        }
15        user = new User();
16        loggedIn = false;
17        return "login?faces-redirect=true";
18    }
19
20    public User getUser() {
21        return user;
22    }
23
24    public boolean isLoggedIn() {
25        return loggedIn;
26    }
27
28    public String logout(){
29        loggedIn = false;
30        return "login?faces-redirect=true";
31    }
32 }
33
```



## MENU.JSP

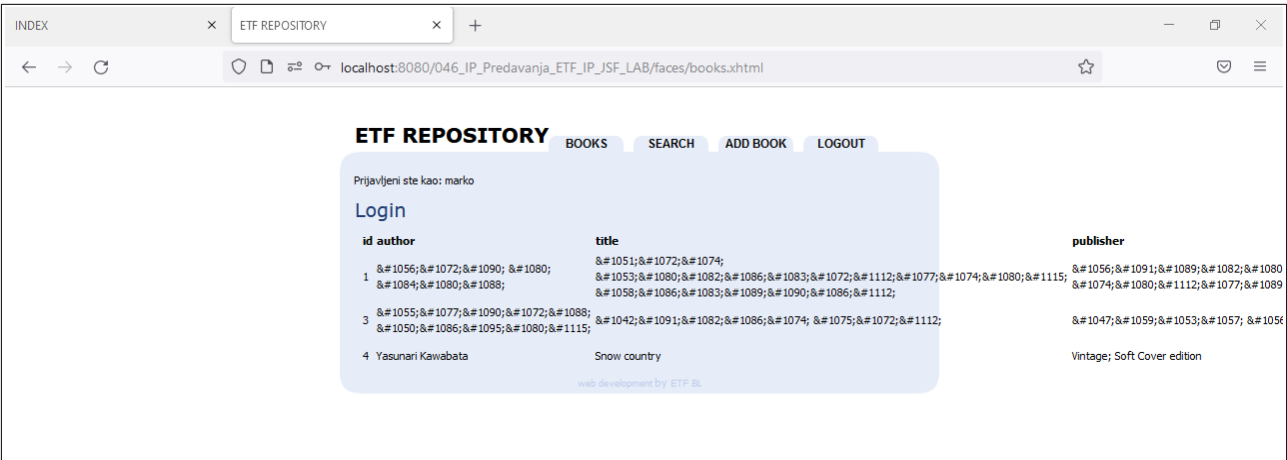
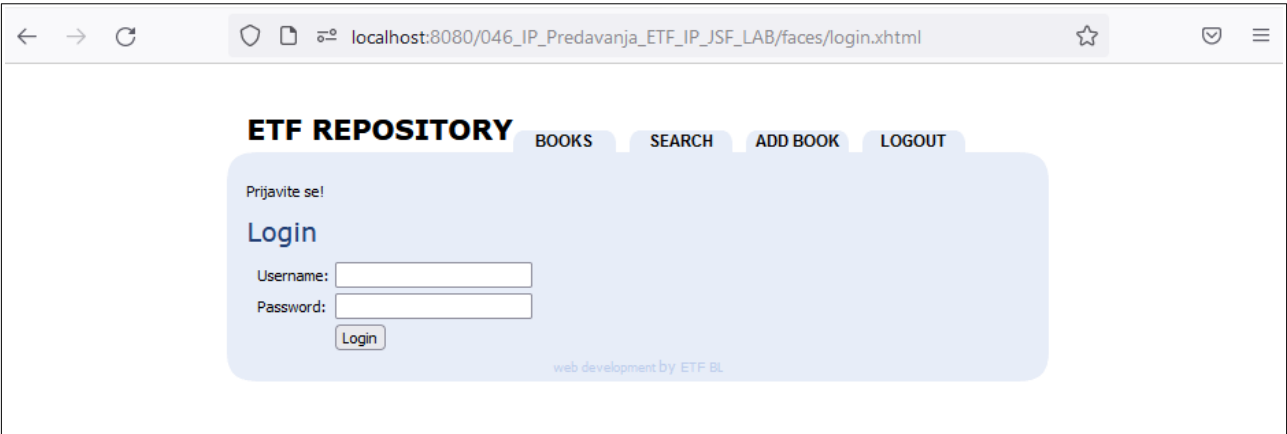
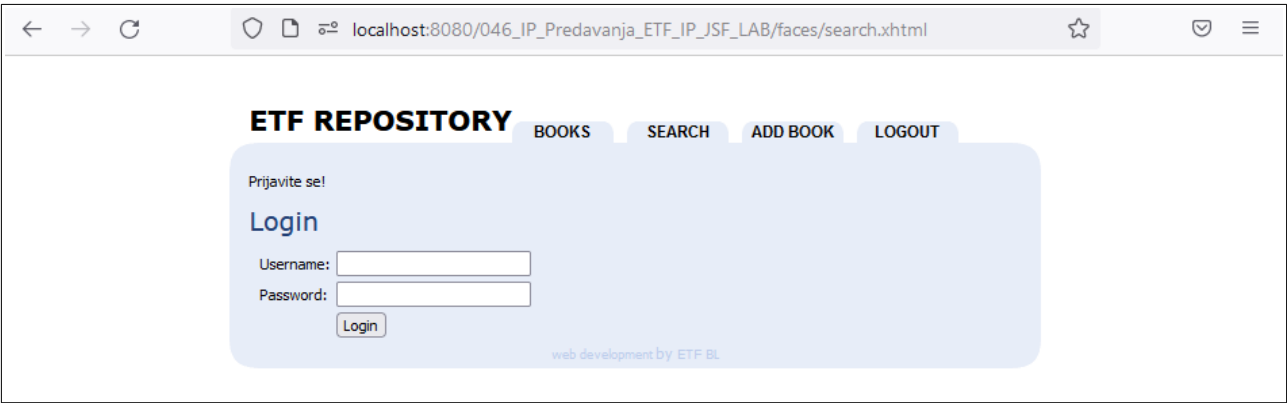
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title>Books Application API Menu</title>
7   </head>
8   <body>
9     <h3>BOOKS APP MENU</h3>
10    <dl>
11      <dd><a href='${pageContext.request.contextPath}/faces/addbook.xhtml'>Add book</a></dd>
12      <dd><a href='${pageContext.request.contextPath}/faces/search.xhtml'>Search books</a></dd>
13      <dd><a href='${pageContext.request.contextPath}/faces/books.xhtml'>Preview books</a></dd>
14      <dd><a href='${pageContext.request.contextPath}/faces/Login.xhtml'>Login/logout form</a></dd>
15    </dl>
16  </body>
17 </html>
```

### BOOKS APP MENU

[Add book](#)  
[Search books](#)  
[Preview books](#)  
[Login/logout form](#)

The screenshot shows a web browser window with the URL `localhost:8080/046_IP_Predavanja_ETF_IP_JSF_LAB/faces/addbook.xhtml`. The page title is "ETF REPOSITORY". The navigation bar includes "BOOKS", "SEARCH", "ADD BOOK", and "LOGOUT". The main content area features a "Login" form with fields for "Username:" and "Password:", and a "Login" button. The text "Prijavite se!" is displayed above the form. The footer text "web development by ETF BL" is visible.

The screenshot shows a web browser window with the URL `localhost:8080/046_IP_Predavanja_ETF_IP_JSF_LAB/faces/books.xhtml`. The page title is "ETF REPOSITORY". The navigation bar includes "BOOKS", "SEARCH", "ADD BOOK", and "LOGOUT". The main content area features a "Login" form with fields for "Username:" and "Password:", and a "Login" button. The text "Prijavite se!" is displayed above the form. The footer text "web development by ETF BL" is visible.



## ETF REPOSITORY

[BOOKS](#)[SEARCH](#)[ADD BOOK](#)[LOGOUT](#)

Prijavljeni ste kao: marko

### Login

Author:	<input type="text"/>
Title:	<input type="text"/>
Publisher:	<input type="text"/>
Year:	<input type="text" value="0"/>
ISBN:	<input type="text"/>
<input type="button" value="Add book"/>	

j\_idt28:author: Validation Error: Value is required.

web development by ETF BL

## ETF REPOSITORY

[BOOKS](#)[SEARCH](#)[ADD BOOK](#)[LOGOUT](#)

Prijavljeni ste kao: marko

### Login

Author:	<input type="text"/>
Title:	<input type="text"/>
Publisher:	<input type="text"/>
Year:	<input type="text" value="0"/>
ISBN:	<input type="text"/>
<input type="button" value="Add book"/>	

j\_idt28:year: Validation Error: Specified attribute is not between the expected values of 1.950 and 2.021.

j\_idt28:isbn: Validation Error: Value is required.

j\_idt28:author: Validation Error: Value is required.

web development by ETF BL

## search.xhtml

```
1<html xmlns="http://www.w3.org/1999/xhtml"
2      xmlns:f="http://java.sun.com/jsf/core"
3      xmlns:ui="http://java.sun.com/jsf/facelets"
4      xmlns:h="http://java.sun.com/jsf/html">
5<h:body>
6<ui:composition template="/WEB-INF/templates/template.xhtml">
7<ui:define name="title">
8    <h:outputText value="ETF REPOSITORY" />
9</ui:define>
10<ui:define name="menu">
11    <ui:include src="/WEB-INF/components/menu.xhtml"></ui:include>
12</ui:define>
13<ui:define name="contentTitle">
14    <h:outputText value="Login" />
15</ui:define>
16<ui:define name="pageContent">
17    <ui:include src="/WEB-INF/components/bodySearchBooks.xhtml"></ui:include>
18</ui:define>
19<ui:define name="footer">
20    <ui:include src="/WEB-INF/components/footer.xhtml"></ui:include>
21</ui:define>
22</ui:composition>
23
24</h:body>
25</html>
```

## login.xhtml

```
1<html xmlns="http://www.w3.org/1999/xhtml"
2      xmlns:f="http://java.sun.com/jsf/core"
3      xmlns:ui="http://java.sun.com/jsf/facelets"
4      xmlns:h="http://java.sun.com/jsf/html">
5<h:body>
6<ui:composition template="/WEB-INF/templates/template.xhtml">
7<ui:define name="title">
8    <h:outputText value="ETF REPOSITORY" />
9</ui:define>
10<ui:define name="menu">
11    <ui:include src="/WEB-INF/components/menu.xhtml"></ui:include>
12</ui:define>
13<ui:define name="contentTitle">
14    <h:outputText value="Login" />
15</ui:define>
16<ui:define name="pageContent">
17    <ui:include src="/WEB-INF/components/bodyLogin.xhtml"></ui:include>
18</ui:define>
19<ui:define name="footer">
20    <ui:include src="/WEB-INF/components/footer.xhtml"></ui:include>
21</ui:define>
22</ui:composition>
23
24</h:body>
25</html>
```

## books.xhtml

```
1<html xmlns="http://www.w3.org/1999/xhtml"
2      xmlns:f="http://java.sun.com/jsf/core"
3      xmlns:ui="http://java.sun.com/jsf/facelets"
4      xmlns:h="http://java.sun.com/jsf/html">
5<h:body>
6<ui:composition template="/WEB-INF/templates/template.xhtml">
7<ui:define name="title">
8    <h:outputText value="ETF REPOSITORY" />
9</ui:define>
10<ui:define name="menu">
11    <ui:include src="/WEB-INF/components/menu.xhtml"></ui:include>
12</ui:define>
13<ui:define name="contentTitle">
14    <h:outputText value="Login" />
15</ui:define>
16<ui:define name="pageContent">
17    <ui:include src="/WEB-INF/components/bodyBooks.xhtml"></ui:include>
18</ui:define>
19<ui:define name="footer">
20    <ui:include src="/WEB-INF/components/footer.xhtml"></ui:include>
21</ui:define>
22</ui:composition>
23
24</h:body>
25</html>
```

## addbook.xhtml

```
1<html xmlns="http://www.w3.org/1999/xhtml"
2      xmlns:f="http://java.sun.com/jsf/core"
3      xmlns:ui="http://java.sun.com/jsf/facelets"
4      xmlns:h="http://java.sun.com/jsf/html">
5<h:body>
6<ui:composition template="/WEB-INF/templates/template.xhtml">
7<ui:define name="title">
8    <h:outputText value="ETF REPOSITORY" />
9</ui:define>
10<ui:define name="menu">
11    <ui:include src="/WEB-INF/components/menu.xhtml"></ui:include>
12</ui:define>
13<ui:define name="contentTitle">
14    <h:outputText value="Login" />
15</ui:define>
16<ui:define name="pageContent">
17    <ui:include src="/WEB-INF/components/bodyAddBook.xhtml"></ui:include>
18</ui:define>
19<ui:define name="footer">
20    <ui:include src="/WEB-INF/components/footer.xhtml"></ui:include>
21</ui:define>
22</ui:composition>
23
24</h:body>
25</html>
```

## template.xhtml

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:f="http://java.sun.com/jsf/core"
4     xmlns:h="http://java.sun.com/jsf/html">
5 <h:head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 <title><ui:insert name="title" /></title>
8 <link rel="stylesheet" type="text/css" href="../css/bluebliss.css" />
9 </h:head>
10
11 <div id="mainContentArea">
12 <div id="contentBox">
13 <div id="title">
14 <ui:insert name="title" />
15 </div>
16
17 <div id="LinkGroup">
18 <ui:insert name="menu" />
19 </div>
20
21 <div id="blueBox">
22 <div id="header"></div>
23 <div class="userLogin">
24 <h:outputText rendered="#{userBean.loggedIn}" value="Prijavljeni ste kao: #{userBean.user.username}" /></div>
25 <h:outputText rendered="#{!userBean.loggedIn}" value="Prijavite se!" /></div>
26 </div>
27 <div class="contentTitle">
28 <ui:insert name="contentTitle" />
29 </div>
30 <div class="pageContent">
31 <ui:insert name="pageContent" />
32 </div>
33 <div id="footer">
34 <ui:insert name="footer" />
35 </div>
36 </div>
37 </div>
38 </div>
39 </html>
```

## menu.xhtml

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:h="http://java.sun.com/jsf/html"
4     xmlns:f="http://java.sun.com/jsf/core">
5 <ui:composition>
6 <h:form>
7 <div class="link"><h:outputLink value="books.xhtml"><h:outputText value="BOOKS"/></h:outputLink></div>
8 <div class="link"><h:outputLink value="search.xhtml"><h:outputText value="SEARCH BOOKS"/></h:outputLink></div>
9 <div class="link"><h:outputLink value="addbook.xhtml"><h:outputText value="ADD BOOK"/></h:outputLink></div>
10 <div class="link"><h:commandLink action="#{userBean.logout}" value="LOGOUT"/></div>
11 </h:form>
12 </ui:composition>
13 </html>
14
```

## footer.xhtml

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2     xmlns:ui="http://java.sun.com/jsf/facelets"
3     xmlns:h="http://java.sun.com/jsf/html"
4     xmlns:f="http://java.sun.com/jsf/core">
5 <ui:composition>
6 <a href="#">web development</a> by <a href="#">ETF BL</a>
7 </ui:composition>
8 </html>
9
```

## bodyAddBook.xhtml

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:f="http://java.sun.com/jsf/core"
5     xmlns:h="http://java.sun.com/jsf/html"
6     xmlns:ui="http://java.sun.com/jsf/facelets">
7 <ui:composition>
8 <h:form>
9 <h:panelGrid columns="2">
10 <h:outputLabel value="Author:"/>
11 <h:inputText id="author" value="#{bookBean.book.author}" required="true">
12 <f:ajax execute="author" render="errorGrid" event="blur"/></f:ajax>
13 </h:inputText>
14 <h:outputLabel value="Title:"/>
15 <h:inputText value="#{bookBean.book.title}"/>
16 <h:outputLabel value="Publisher:"/>
17 <h:inputText value="#{bookBean.book.publisher}"/>
18 <h:outputLabel value="Year:"/>
19 <h:inputText id="year" value="#{bookBean.book.year}>
20 <f:validateLongRange minimum="1950" maximum="2021"/>
21 </h:inputText>
22 <h:outputLabel value="ISBN:"/>
23 <h:inputText id="isbn" value="#{bookBean.book.isbn}" required="true"/>
24 <h:outputLabel/>
25 <h:commandButton action="#{bookBean.add}" value="Add book"/>
26
27 </h:panelGrid>
28 <h:panelGrid columns="2" id="errorGrid">
29 <h:outputLabel/>
30 <h:message for="year"/>
31 <h:outputLabel/>
32 <h:message for="isbn"/>
33 <h:outputLabel/>
34 <h:message for="author"/>
35 </h:panelGrid>
36 </h:form>
37 </ui:composition>
38
39 </html>
```

## bookLogin.xhtml

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:f="http://java.sun.com/jsf/core"
5     xmlns:h="http://java.sun.com/jsf/html"
6     xmlns:ui="http://java.sun.com/jsf/facelets"
7     xmlns:c="http://java.sun.com/jsp/jstl/core"
8     xmlns:o="http://omnifaces.org/ui"
9     xmlns:of="http://omnifaces.org/functions">
10
11 <ui:composition>
12 <h:form>
13 <h:panelGrid columns="2">
14 <h:outputLabel value="Username:"/>
15 <h:inputText value="#{userBean.user.username}"/>
16 <h:outputLabel value="Password:"/>
17 <h:inputSecret value="#{userBean.user.password}"/>
18 <h:outputLabel/>
19 <h:commandButton action="#{userBean.login}" value="Login"/>
20 </h:panelGrid>
21 </h:form>
22
23
24 </ui:composition>
25
26 </html>
```

## bodyBooks.xhtml

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:f="http://java.sun.com/jsf/core"
5       xmlns:h="http://java.sun.com/jsf/html"
6       xmlns:ui="http://java.sun.com/jsf/facelets">
7 <ui:composition>
8   <h:form>
9       <h:dataTable id="booksTable" value="#{bookBean.all}" var="book">
10           <h:column>
11               <f:facet name="header">
12                   <h:outputText value="id" />
13               </f:facet>
14               <h:outputText value="#{book.id}" />
15           </h:column>
16           <h:column>
17               <f:facet name="header">
18                   <h:outputText value="author" />
19               </f:facet>
20               <h:outputText value="#{book.author}" />
21           </h:column>
22           <h:column>
23               <f:facet name="header">
24                   <h:outputText value="title" />
25               </f:facet>
26               <h:outputText value="#{book.title}" />
27           </h:column>
28           <h:column>
29               <f:facet name="header">
30                   <h:outputText value="publisher" />
31               </f:facet>
32               <h:outputText value="#{book.publisher}" />
33           </h:column>
34           <h:column>
35               <f:facet name="header">
36                   <h:outputText value="isbn" />
37               </f:facet>
38               <h:outputText value="#{book.isbn}" />
39           </h:column>
40           <h:column>
41               <f:facet name="header">
42                   <h:outputText value="action" />
43               </f:facet>
44               <h:commandLink action="#{bookBean.delete}">
45                   <f:setPropertyActionListener target="#{bookBean.book}" value="#{book}" />
46                   <h:outputText title="Obriši" value="Obriši" />
47               </h:commandLink>
48           </h:column>
49       </h:dataTable>
50   </h:form>
51 </ui:composition>
52
53 </html>
```



## bodySearchBooks.xhtml

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:f="http://java.sun.com/jsf/core"
5       xmlns:h="http://java.sun.com/jsf/html"
6       xmlns:ui="http://java.sun.com/jsf/facelets">
7 <ui:composition>
8   <h:form>
9       <h:inputText id="search" value="#{bookBean.searchText}">
10         <f:ajax execute="search" render="booksTable" event="keyup"/></f:ajax>
11       </h:inputText>
12       <h:dataTable id="booksTable" value="#{bookBean.booksByTitle}" var="book">
13         <h:column>
14           <f:facet name="header">
15             <h:outputText value="id" />
16           </f:facet>
17           <h:outputText value="#{book.id}" />
18         </h:column>
19         <h:column>
20           <f:facet name="header">
21             <h:outputText value="author" />
22           </f:facet>
23           <h:outputText value="#{book.author}" />
24         </h:column>
25         <h:column>
26           <f:facet name="header">
27             <h:outputText value="title" />
28           </f:facet>
29           <h:outputText value="#{book.title}" />
30         </h:column>
31         <h:column>
32           <f:facet name="header">
33             <h:outputText value="publisher" />
34           </f:facet>
35           <h:outputText value="#{book.publisher}" />
36         </h:column>
37         <h:column>
38           <f:facet name="header">
39             <h:outputText value="isbn" />
40           </f:facet>
41           <h:outputText value="#{book.isbn}" />
42         </h:column>
43         <h:column>
44           <f:facet name="header">
45             <h:outputText value="action" />
46           </f:facet>
47           <h:commandLink action="#{bookBean.delete}">
48             <f:setPropertyActionListener target="#{bookBean.book}" value="#{book}" />
49             <h:outputText title="Obriši" value="Obriši" />
50           </h:commandLink>
51         </h:column>
52       </h:dataTable>
53     </h:form>
54 </ui:composition>
55
56 </html>
```

