МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Тольяттинский государственный университет»

|  |
| --- |
| *(наименование института полностью)* |
|  |
| (*Наименование учебного структурного подразделения*) |
| 09.03.03 Прикладная информатика |
| *(код и наименование направления подготовки / специальности)* |
| Разработка социальных и экономических информационных систем |
| *(направленность (профиль) / специализация)* |

**Практическое задание №3**

* по учебному курсу «[Технологии и средства конструирования программного обеспечения](https://edu.rosdistant.ru/course/view.php?id=11365" \o "Технологии и средства конструирования программного обеспечения)»

|  |  |  |
| --- | --- | --- |
| Обучающегося | Елшина И.А. |  |
|  |  |  |
| Группа | ПИбп-2004а |  |
|  |  |  |
| Преподаватель | Гущина О.М. |  |
|  | *(И.О. Фамилия)* |  |

Тольятти 2025

**Цель работы**

Сформировать навыки разработки и реализации веб-приложения.

**Задание.** Разработка веб-приложения. Создание простейшего сервлета.

**Задание.** Разработка веб-приложения.

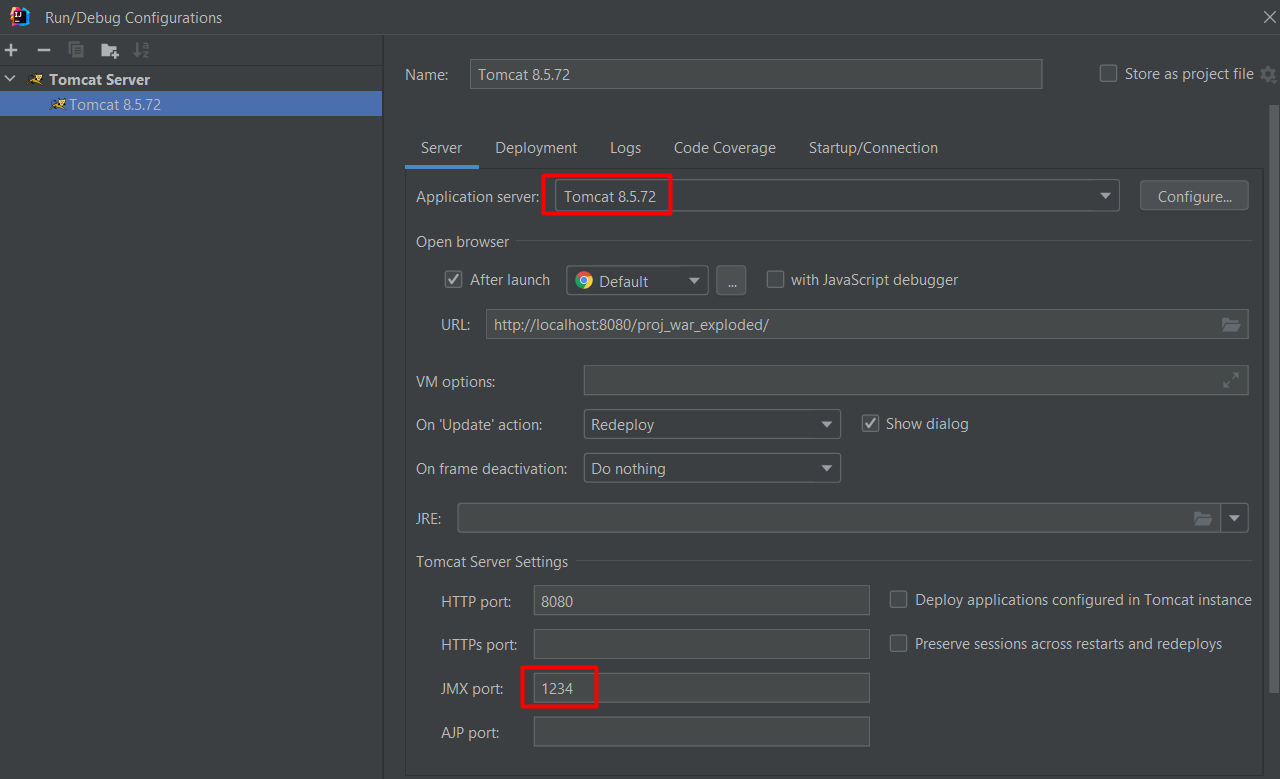
|  |  |
| --- | --- |
| № варианта | Предметная область |
| 2 | Аптечный склад |

Целью данной научно-исследовательской работы являлась разработка концептуального проекта корпоративной информационной системы для автоматизации бизнес-процессов на предприятии оптовой торговли фармацевтическими препаратами.

В качестве базовой архитектуры было выбрано трехзвенное клиент-серверное веб-приложение на платформе Java Enterprise Edition с возможностью масштабирования.

Ядром серверной части выступает веб-контейнер Apache Tomcat, обеспечивающий выполнение Java-сервлетов и JSP-страниц, формирующих динамический HTML-контент.

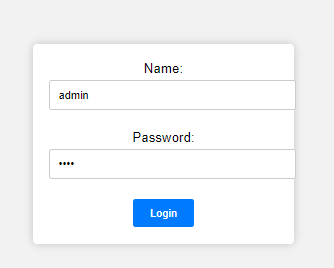
Разработка осуществлялась в среде NetBeans IDE с использованием возможностей автоматической сборки и деплоя веб-приложений на сервер приложений.

Ниже показаны параметры запуска.

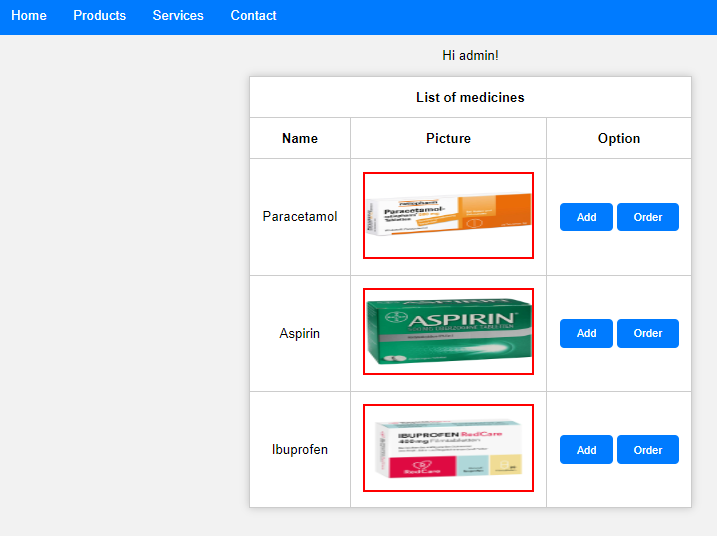
Основными подсистемами проектируемой ИС являются:

* Подсистема складского учета и логистики
* Подсистема управленческого учета и отчетности
* Подсистема электронного документооборота
* Подсистема безопасности и разграничения доступа

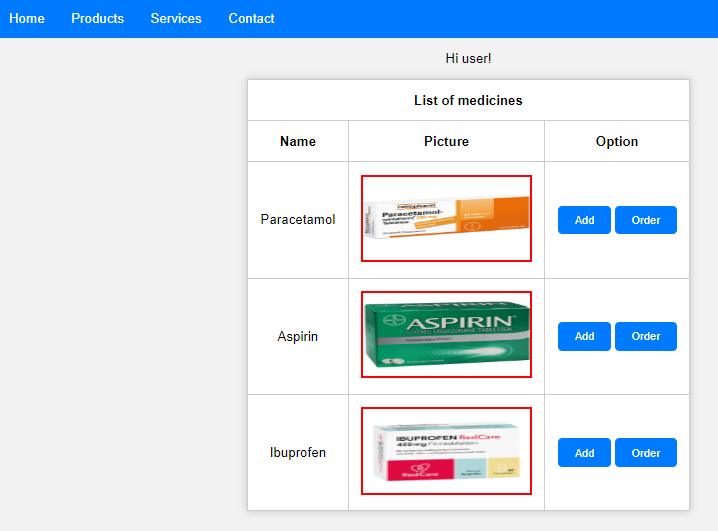
Важной особенностью решения стала концепция единого хранилища JSP-шаблонов и библиотеки тегов для унификации пользовательского интерфейса подсистем.



Окно для администратора:



Окно для пользователя:



В данном случае .jspf файл сохранен по пути webapp/WEB-INF/jspf/header.jspf и представляет из себя код для добавления навменю с стилями в верхушку страницы.

Результатом работы стал прототип кроссплатформенной ИС, демонстрирующий архитектуру и ключевой функционал системы с возможностью дальнейшей доработки и масштабирования решения.

**Выводы**

В процессе выполнения научно-исследовательской работы был разработан прототип кроссплатформенного веб-приложения для автоматизации бизнес-процессов в предметной области "Аптечный склад" с применением инструментария и технологий корпоративной платформы Java Enterprise Edition.

В качестве серверных технологий, формирующих бизнес-логику веб-приложения на стороне сервера, были выбраны Java-сервлеты - спецификация для разработки многопоточных веб-приложений, взаимодействующих с клиентами по протоколу HTTP, а также технология JavaServer Pages (JSP) - подход к созданию динамических веб-страниц на основе HTML и набора реузабельных компонентов.

Для локального развёртывания и тестирования разработанного веб-приложения использовался программный комплекс Apache Tomcat, реализующий серверные технологии Java Servlet и JavaServer Pages, а также предоставляющий полноценную среду выполнения для JSP и Java-классов.

В результате проделанной работы был создан работоспособный прототип распределённой корпоративной информационной системы, демонстрирующий возможности выбранных технологий Java EE для разработки веб-ориентированных бизнес-приложений. Дальнейшее совершенствование системы может включать масштабирование, повышение отказоустойчивости, оптимизацию архитектуры, а также расширение бизнес-функциональности.

**Листинг программы**

HelloServlet.java

package com.example.proj;  
  
import java.io.\*;  
import javax.servlet.RequestDispatcher;  
import javax.servlet.ServletException;  
import javax.servlet.http.\*;  
import javax.servlet.annotation.\*;  
  
//Сайт для аптеки.  
// логин для аптекаря - admin/pass  
// логин для пользователя - user/user\_pass  
@WebServlet(name = "helloServlet", value = "/hello-servlet")

public class HelloServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Получаем параметры из запроса: логин и пароль

String n = request.getParameter("username");

String p = request.getParameter("userpass");

// Проверяем валидность логина и пароля для администратора и пользователя

ConnectionStatus adminStatus = LoginDao.validateAdmin(n, p);

ConnectionStatus userStatus = LoginDao.validateUser(n, p);

if (adminStatus == ConnectionStatus.WRONG\_PASSWORD || userStatus == ConnectionStatus.WRONG\_PASSWORD) {

// Если пароль неверный для администратора или пользователя, перенаправляем на страницу с ошибкой

RequestDispatcher rd = request.getRequestDispatcher("wrong-login-servlet");

rd.forward(request, response);

} else if (adminStatus == ConnectionStatus.OK) {

// Если администратор успешно вошел, перенаправляем на админскую страницу

RequestDispatcher rd = request.getRequestDispatcher("admin-servlet");

rd.forward(request, response);

} else if (userStatus == ConnectionStatus.OK) {

// Если пользователь успешно вошел, перенаправляем на приветственную страницу

RequestDispatcher rd = request.getRequestDispatcher("welcome-servlet");

rd.forward(request, response);

} else {

// Если ни один логин не соответствует, выводим сообщение об ошибке и перенаправляем на главную страницу

out.print("No such login exists");

RequestDispatcher rd = request.getRequestDispatcher("index.html");

rd.include(request, response);

}

out.close();

}

public void destroy() {

// Дополнительные действия при завершении работы сервлета (по желанию)

}

}

WrongLoginServlet.java

package com.example.proj;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.io.PrintWriter;  
  
@WebServlet(name = "wrongLoginServlet", value = "/wrong-login-servlet")

public class WrongLoginServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Устанавливаем тип контента в ответе как HTML

response.setContentType("text/html");

// Получаем объект для записи в ответ

PrintWriter out = response.getWriter();

// Выводим сообщение об ошибке входа

out.print("Wrong login!\n");

// Закрываем поток

out.close();

}

}

WelcomeServlet.java

package com.example.proj;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
@WebServlet(name = "welcomeServlet", value = "/welcome-servlet")  
public class WelcomeServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Устанавливаем тип контента в ответе как HTML

response.setContentType("text/html");

// Получаем объект для записи в ответ

PrintWriter out = response.getWriter();

// Генерируем HTML-страницу

out.print("<!DOCTYPE html> \n" +

" <html>\n");

// Выводим приветственное сообщение для пользователя

out.print("<hr>Welcome back, user! Which medicine would you like to buy?</hr>");

// Вставляем кнопки с надписью "Buy" с использованием метода из класса Pictures

out.println(Pictures.buttonName("Buy"));

// Закрываем HTML-страницу

out.print("</html>\n");

// Закрываем поток

out.close();

}

}

Pictures.java

package com.example.proj;  
  
public class Pictures {

// Метод для генерации HTML-кода таблицы с кнопками

public static String buttonName(String name){

return "<table border=\"3\" bordercolor=\"red\" align=\"center\">\n" +

" <tr>\n" +

" <th colspan=\"3\">List of medicines</th> \n" +

" </tr>\n" +

" <tr>\n" +

" <th>Name</th>\n" +

" <th>Picture</th>\n" +

" <th>Option</th>\n" +

" </tr>\n" +

" <tr>\n" +

" <td>Paracetamol</td>\n" + // Закрытие тега должно быть </td> вместо </th>

" <td><img src=\"WebContent\\par.jpg\" border=3 height=100 width=200></img></td>\n" +

" <td><button type=\"button\">" + name + "</button>" +

"<button type=\"button\">Process order</button></td>\n" + // Закрытие тега должно быть </td> вместо </th>

" </tr>\n" +

" <tr>\n" +

" <td>Aspirin</td>\n" + // Закрытие тега должно быть </td> вместо </th>

" <td><img src=\"WebContent\\asp.jpg\" border=3 height=100 width=200></img></td>\n" +

" <td><button type=\"button\">" + name + "</button>" +

"<button type=\"button\">Process order</button></td>\n" + // Закрытие тега должно быть </td> вместо </th>

" </tr>\n" +

" <tr>\n" +

" <td>Ibuprofen</td>\n" + // Закрытие тега должно быть </td> вместо </th>

" <td><img src=\"WebContent\\ibu.jpg\" border=3 height=100 width=200></img></td>\n" +

" <td><button type=\"button\">" + name + "</button>" +

" <button type=\"button\">Process order</button></td>\n" + // Закрытие тега должно быть </td> вместо </th>

" </tr>\n" +

"</table>\n";

}

}

LoginDao.java

package com.example.proj;  
  
public class LoginDao {  
  
public static ConnectionStatus validateAdmin(String name, String pass) {

// Проверяем, является ли имя "admin" и пароль "pass" правильными

if (name.equals("admin") && pass.equals("pass"))

return ConnectionStatus.OK;

// Если имя "admin", но пароль неверный, возвращаем статус "WRONG\_PASSWORD"

if (name.equals("admin"))

return ConnectionStatus.WRONG\_PASSWORD;

// Если имя не "admin", возвращаем статус "NO"

return ConnectionStatus.NO;

}

public static ConnectionStatus validateUser(String name, String pass) {

// Проверяем, является ли имя "user" и пароль "user\_pass" правильными

if (name.equals("user") && pass.equals("user\_pass"))

return ConnectionStatus.OK;

// Если имя "user", но пароль неверный, возвращаем статус "WRONG\_PASSWORD"

if (name.equals("user"))

return ConnectionStatus.WRONG\_PASSWORD;

// Если имя не "user", возвращаем статус "NO"

return ConnectionStatus.NO;

}

ConnectionStatus.java

package com.example.proj;  
  
public enum ConnectionStatus {  
 *OK*, *WRONG\_PASSWORD*, *NO*}

AdminServlet.java

package com.example.proj;  
  
import java.io.IOException;  
import java.io.PrintWriter;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
  
@WebServlet(name = "adminServlet", value = "/admin-servlet")

public class AdminServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// Устанавливаем тип контента в ответе как HTML

response.setContentType("text/html");

// Получаем объект для записи в ответ

PrintWriter out = response.getWriter();

// Генерируем HTML-страницу

out.print("<!DOCTYPE html> \n" +

" <html>\n");

// Выводим приветственное сообщение для администратора

out.print("<hr>Welcome back, admin! Which medicine is to add today?</hr>");

// Вставляем кнопки с надписью "Add" с использованием метода из класса Pictures

out.println(Pictures.buttonName("Add"));

// Закрываем HTML-страницу

out.print("</html>\n");

// Закрываем поток

out.close();

}

}

header.jspf

<%@ **page** contentType="text/html; charset=UTF-8" language="java" %>  
<%@ **page** contentType="text/html; charset=UTF-8" %>  
  
<!-- Навигационное меню со стилями из HTML-кода -->  
<ul class="navbar">  
 <li><a href="#">Home</a></li>  
 <li><a href="#">Products</a></li>  
 <li><a href="#">Services</a></li>  
 <li><a href="#">Contact</a></li>  
</ul>