

Рно Федеральное агентство по образованию
Государственное образовательное учреждение
высшего профессионального образования
Тульский государственный университет
Кафедра автоматики и телемеханики

Работа с базами данных в системе Dreamweaver

**Методические указания к лабораторной работе
по курсу «Сетевое программирование»**

Тула 2010

Цель работы. Создать веб-приложение, осуществляющее работу с базой данных. Для добавления динамики на веб-страницах использовать Java Script.

1. Теоретическая справка

Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами.

Для разработки и тестирования динамических веб-страниц (окончательное содержимое страницы зависит от запроса, созданного на основе действий посетителя) необходимо наличие работающего *веб-сервера* — это программное обеспечение, которое предоставляет веб-страницы в ответ на запросы веб-браузеров. Например, Microsoft Internet Information Server (IIS) и Apache HTTP Server.

Сервер приложений — программное обеспечение, с помощью которого веб-сервер обрабатывает динамические страницы. Выбор сервера приложений должен производиться с учетом применяемой серверной технологии и типа веб-сервера.

В таблице 1 приведены наиболее распространенные серверы приложений, которые доступны для серверных технологий, поддерживаемых DreamWeaver.

Таблица 1

Серверная технология	Сервер приложений
ColdFusion	Adobe ColdFusion 8
ASP	Сервер Microsoft IIS
PHP	Сервер PHP
JSP	Jaguar CTS

2. Клиентская часть разработки приложения

Рассмотрим разработку веб-приложения на примере работы с гостевой книгой.

Первая стадия разработки веб-приложения – это создание HTML страниц с подключением каскадных таблиц стилей (CSS) и функций на языке Java Script для динамического отображения данных и проверки ввода пользователя, выполняющихся в браузере пользователя, т.е. на стороне клиента.

Язык JavaScript - это объектно-ориентированный язык, предназначенный для создания приложений в Интернете. Язык JavaScript является системно-независимым и совместим между всеми типами компьютеров, работающими в Интернет. Программы на языке JavaScript включаются в состав HTML документа и, фактически, этот язык может считаться расширением состава команд HTML. Поэтому для создания программ на JavaScript не требуется никаких дополнительных средств, необходим лишь браузер, поддерживающий JavaScript.

Как и CSS, функции, написанные на JavaScript можно хранить в отдельном файле с расширением *.js, который подключается ко всем страницам, которые эти функции используют. Ссылка на такой файл с именем main.js в гипертекстовом документе выглядит следующим образом:

```
<SCRIPT language=JavaScript1.2 src="main.js"></SCRIPT>
```

Сам файл main.js содержит в данном веб-приложении две функции: fulltime() и FTest (). Первая выводит на экран время в формате «ДД месяц ГГГГ ЧЧ ММ СС», причем время обновляется каждую секунду. Вторая функция необходима для тестирования. В листинге 2 приведен текст функции fulltime() файла main.js:

Листинг 2

```
// JavaScript Document
function fulltime() {
    var time=new Date();
    document.clock.full.value=time.toLocaleString();
    setTimeout('fulltime()',500)
}
```

Эти функции вызываются из каждой страницы сайта таким образом:

```
<script language="JavaScript"> fulltime(); </script>
```

Еще одна функция JavaScript используется в приложении для проверки, ввел ли посетитель страницы имя и сообщение в форму гостевой книги. Код этой функции размещается в том же документе, что и соответствующая форма. В листинге 3 представлен текст функции, осуществляющей проверку:

Листинг 3

```
<script language = "JavaScript">
function check_1()
{
    if (document.frml.nm.value == "" || document.frml.text.value ==
"")
        {alert("Заполните все поля формы!");
return (false);}
    else
        return (true);
}
</script>
```

Если хотя бы одно из полей ввода формы осталось незаполненным, то при помощи метода alert() пользователь информируется об этом:

```
alert("Заполните все поля формы!");
```

Функция check_1() вызывается из обработчика события onSubmit() формы таким образом:

```
<form name = frml method="POST" action="2.jsp"
onSubmit="return(check_1())">.
```

Если функция возвращает false, то передача данных на сервер не происходит.

На рисунке 1 представлена одна из страниц приложения, открытая в браузере и демонстрирующая применение стилей и скриптов:

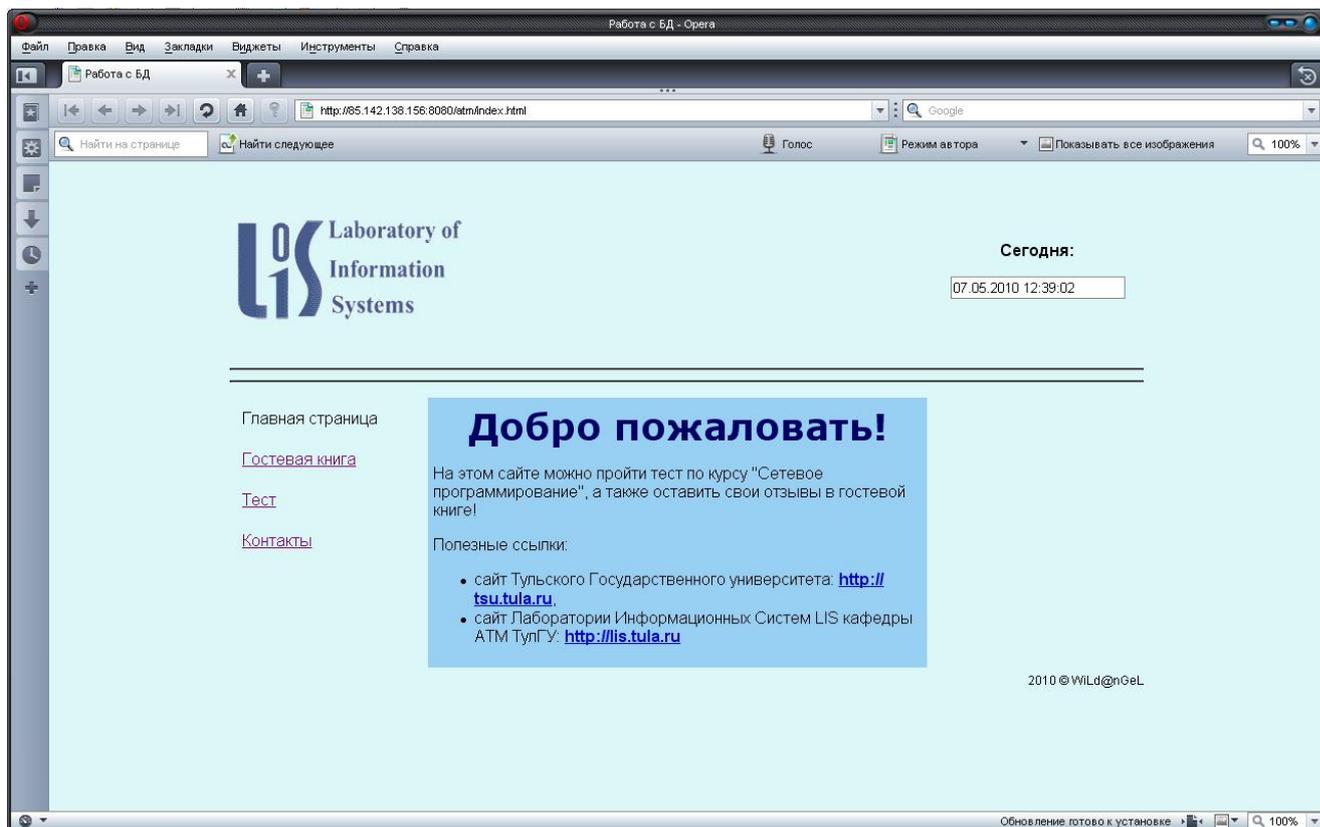


Рисунок 1 – Страница приложения, демонстрирующая применение стилей и скриптов

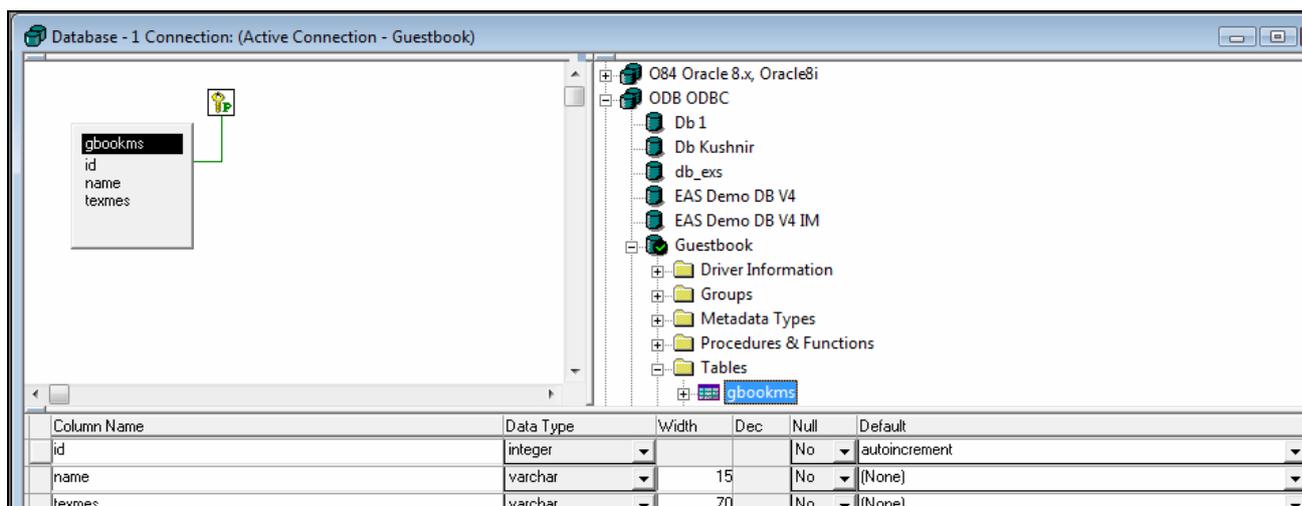
3. Серверная часть разработки приложения

Для создания серверной части приложения используется технология Java Server Pages (JSP). Технология Java Server Pages (JSP) является составной частью единой технологии создания бизнес-приложений J2EE. JSP - это альтернативная методика разработки приложений, динамически генерирующих ответ на те или иные запросы клиента.

Для тестирования работоспособности приложения будем использовать web-сервер Jaguar CTS и сервер баз данных Adaptive Server Anywhere компании Sybase.

Создадим базу данных guestbook.db в PowerBuilder, содержащую таблицу для хранения записей гостевой книги gbookms. Таблица состоит из трех колонок: id, name, texmes, соответственно для хранения идентификационного номера сообщения (поле является автоинкрементным, первичным ключом таблицы),

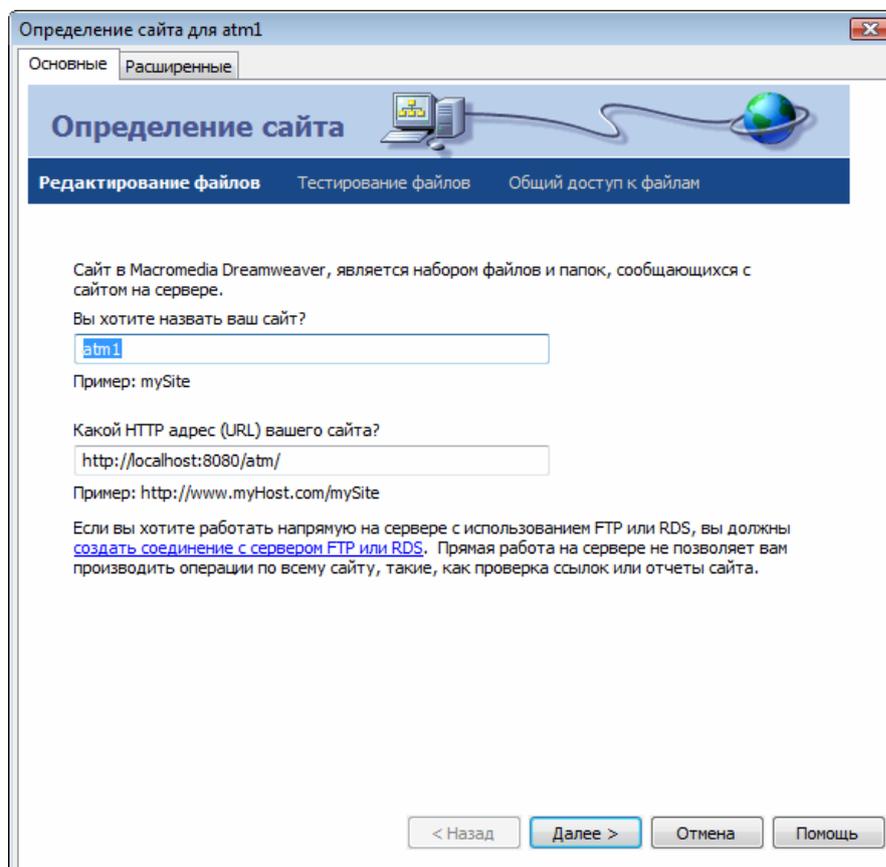
имени пользователя и текста сообщения. Структура таблицы показана на рисунке 2.



Column Name	Data Type	Width	Dec	Null	Default
id	integer			No	autoincrement
name	varchar	15		No	(None)
texmes	varchar	70		No	(None)

Рисунок 2 – Таблица базы данных для хранения сообщений гостевой книги

В системе DreamWeaver создадим настройки для сайта. Для этого выберем пункт главного меню «Сайт -> Новый сайт». В открывшемся мастере создания сайтов пропишем настройки для web-сервера Jaguar CTS, как показано на рисунках 3 – 8:



Определение сайта для atm1

Основные | Расширенные

Определение сайта

Редактирование файлов | Тестирование файлов | Общий доступ к файлам

Сайт в Macromedia Dreamweaver, является набором файлов и папок, сообщающихся с сайтом на сервере.

Вы хотите назвать ваш сайт?

Пример: mySite

Какой HTTP адрес (URL) вашего сайта?

Пример: http://www.myHost.com/mySite

Если вы хотите работать напрямую на сервере с использованием FTP или RDS, вы должны [создать соединение с сервером FTP или RDS](#). Прямая работа на сервере не позволяет вам производить операции по всему сайту, такие, как проверка ссылок или отчеты сайта.

< Назад | Далее > | Отмена | Помощь

Рисунок 3 – Создание настроек для сайта, шаг 1

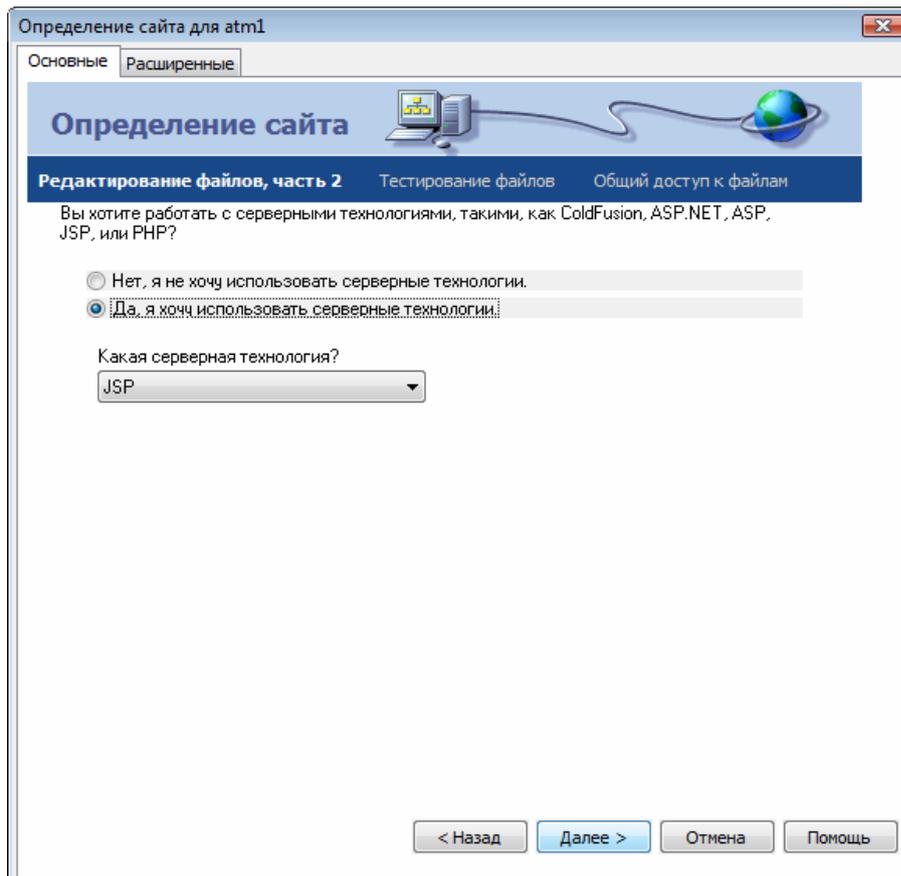


Рисунок 4 – Создание настроек для сайта, шаг 2

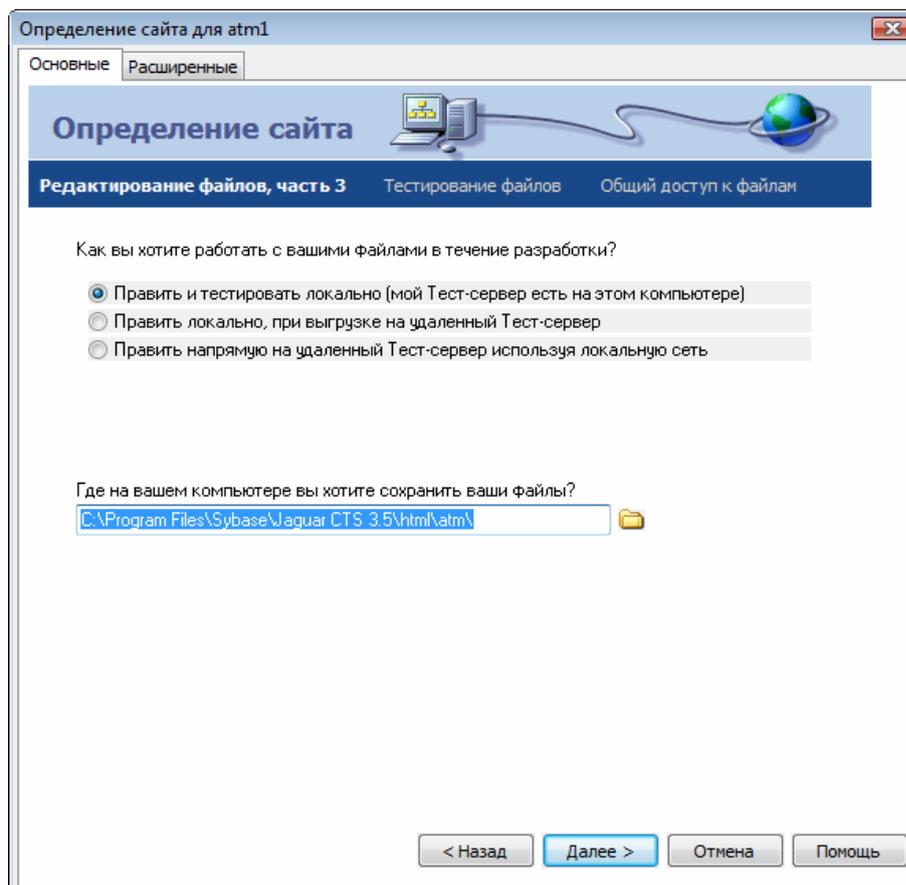


Рисунок 5 – Создание настроек для сайта, шаг 3

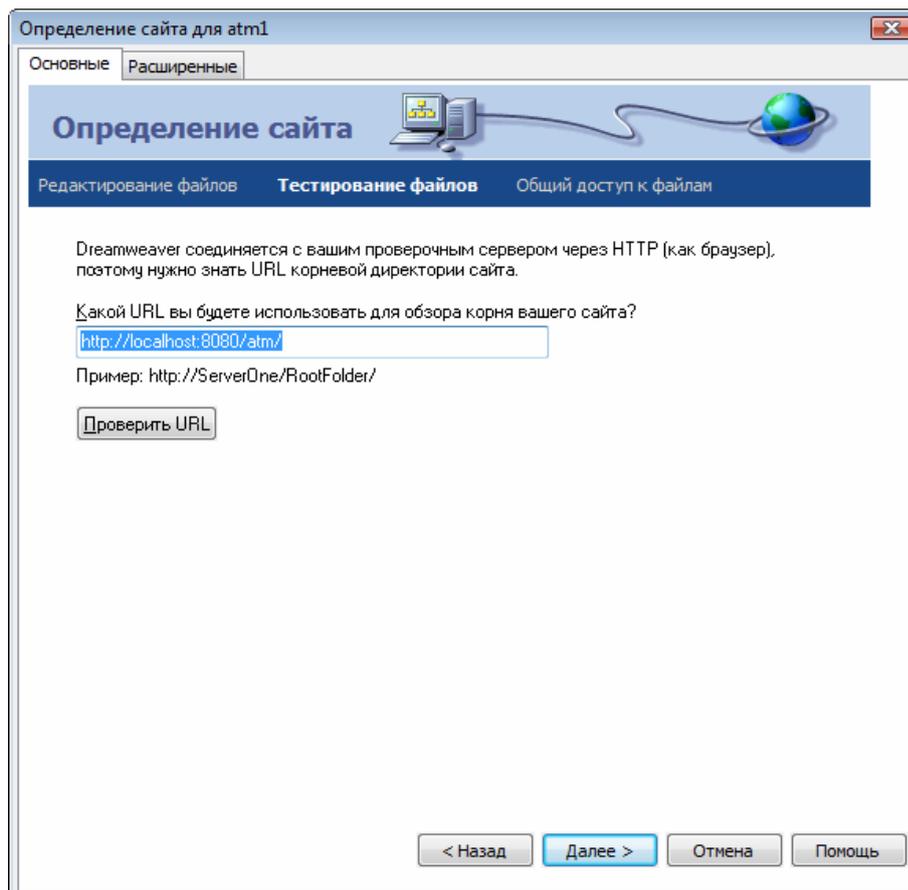


Рисунок 6 – Создание настроек для сайта, шаг 4

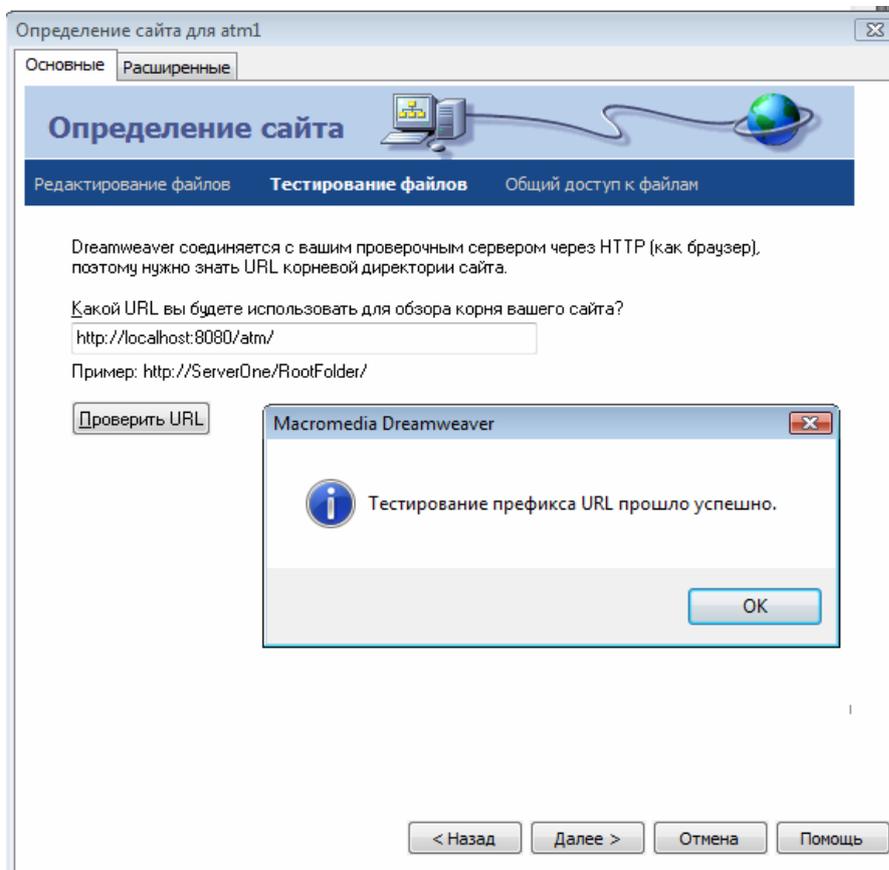


Рисунок 7 – Тестирование сервера

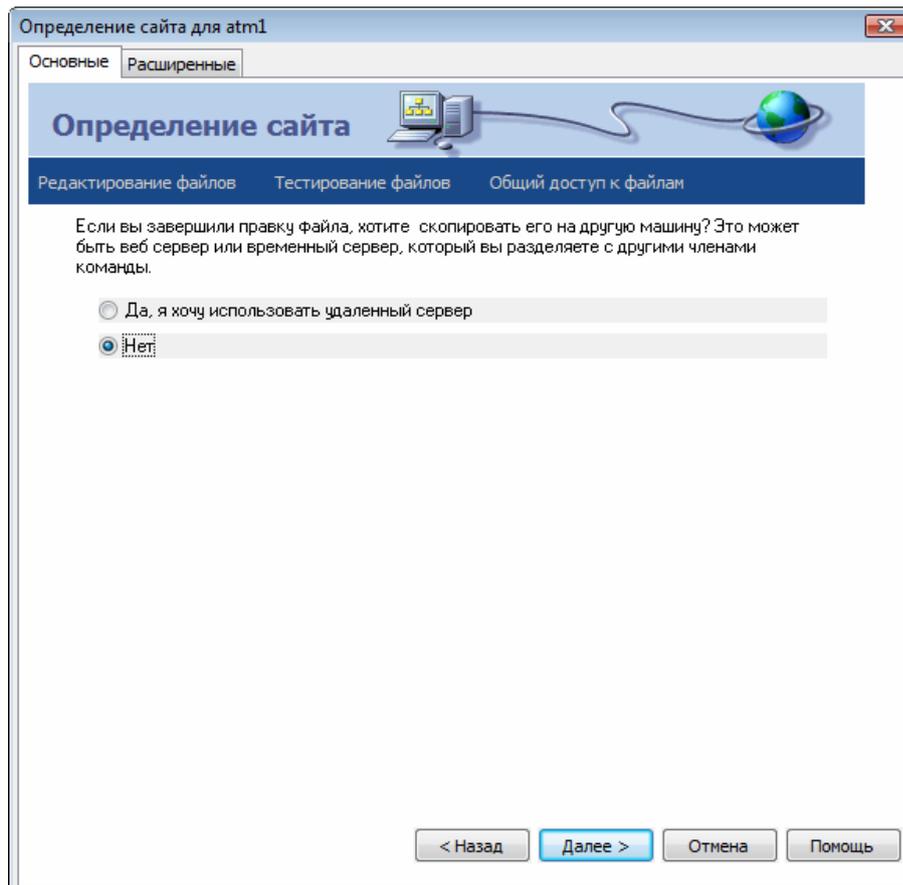
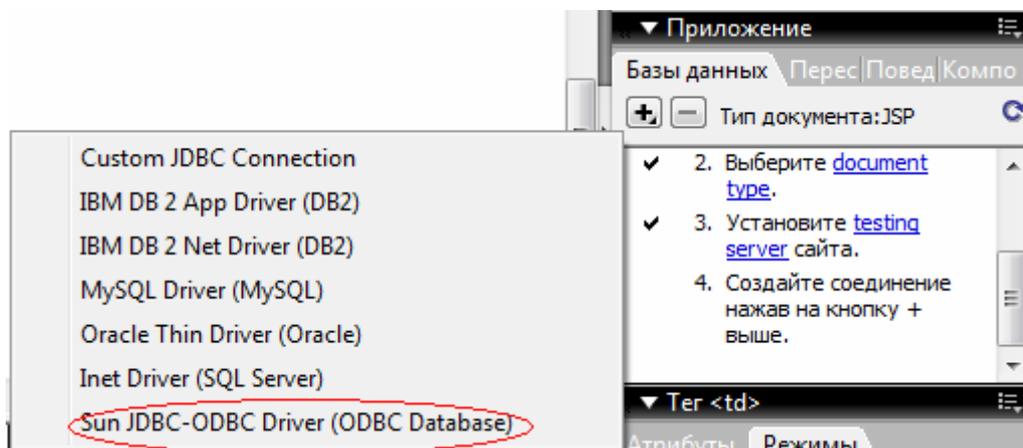


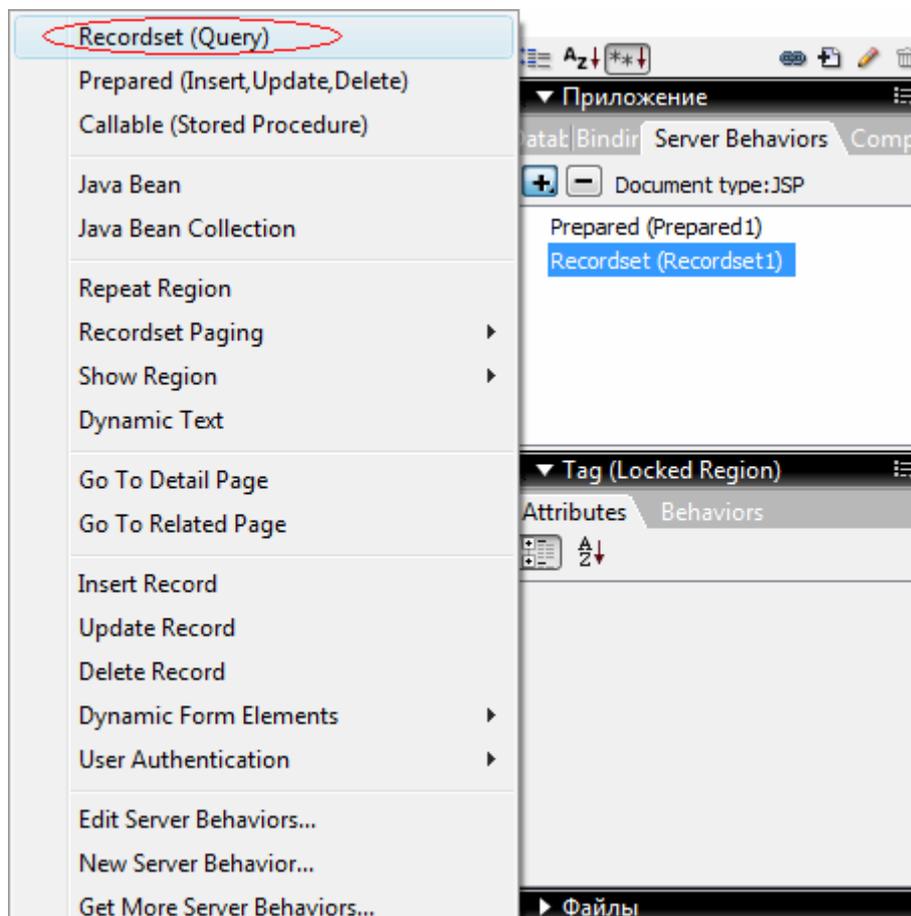
Рисунок 8 – Заключительные настройки

Создадим соединение с базой данных при помощи панели «**Приложение**» системы DreamWeaver. На вкладке «**Базы данных**» («**Databases**») выберем подходящий драйвер для работы с базой данных, как показано на рисунке 9. Для базы данных Adaptive Server Anywhere компании Sybase выбираем драйвер **Sun JDBC-ODBC (ODBC Database)**.



a)

11а, в окне создания запроса сконструировать запрос **SELECT** к базе данных. В результате DreamWeaver вставит JSP-код в страницу, как показано на рисунке 11б.



a)

```

21 Prepared1.executeUpdate();
22 %>
23 <%
24 Driver DriverRecordset1 = (Driver)Class.forName(MM_gbook_DRIVER).newInstance();
25 Connection ConnRecordset1 = DriverManager.getConnection(MM_gbook_STRING,MM_gbook_USERNAME,MM_gbook_PASSWORD);
26 PreparedStatement StatementRecordset1 = ConnRecordset1.prepareStatement("SELECT * FROM DBA.gbookms");
27 ResultSet Recordset1 = StatementRecordset1.executeQuery();
28 boolean Recordset1_isEmpty = !Recordset1.next();
29 boolean Recordset1_hasData = !Recordset1_isEmpty;
30 Object Recordset1_data;
31 int Recordset1_numRows = 0;
32 %>
33 < %@page import="java.sql.*"%>

```

b)

Рисунок 11 – JSP-код для выбора всех записей из базы данных, созданный системой DreamWeaver

Чтобы вывести полученные записи на станицу, добавим код, показанный в листинге 4:

Листинг 4

```
<%
```

```

        out.print("ID: "+Recordset1.getString("id")+"&nbsp;&nbsp; ");
        out.print("Name: "+Recordset1.getString("name")+"<br>");
        out.print("Message: "+Recordset1.getString("texmes")+"<br>");
        out.print("<hr>");

%>
<%
    while(Recordset1.next())
    {
        out.print("ID: "+Recordset1.getString("id")+"&nbsp;&nbsp; ");
        out.print("Name: "+Recordset1.getString("name")+"<br>");
        out.print("Message: "+Recordset1.getString("texmes")+"<br>");
        out.print("<hr>");
    }
%>

```

Теперь создадим возможность добавления записей из гостевой книги в базу данных. Сначала надо получить введенные пользователем в форму данные. Для этого создаются переменные `name1` и `text1` и инициализируются при помощи метода **getParameter()**, как показано в листинге 5:

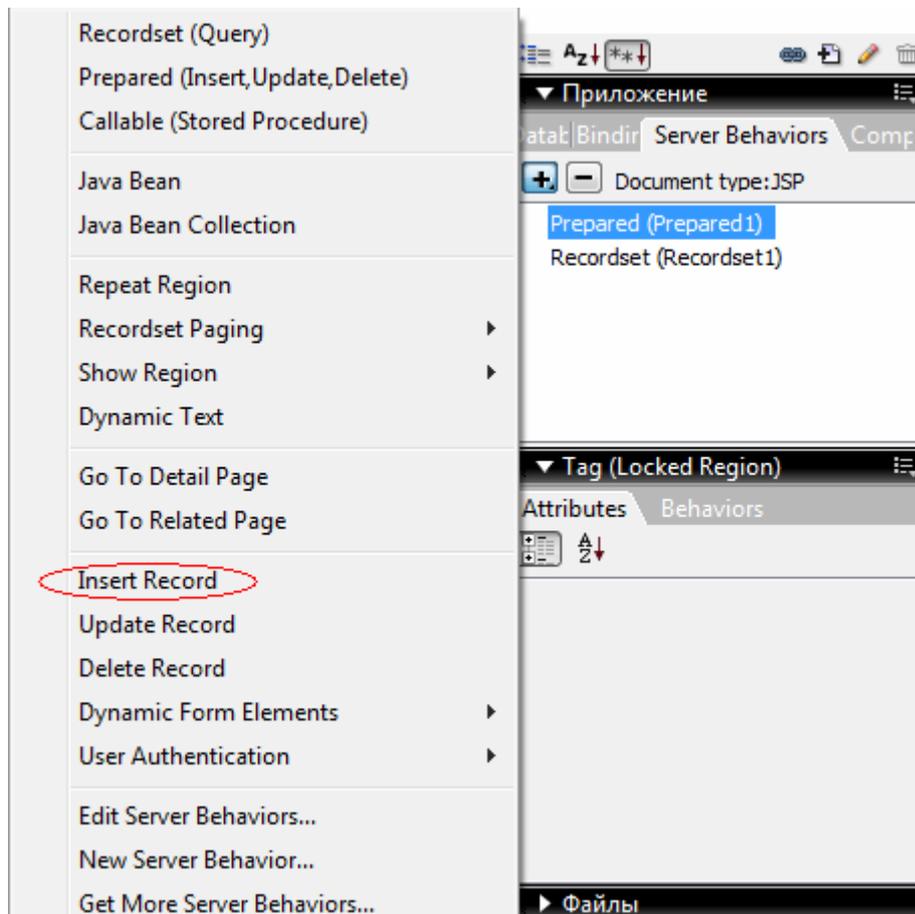
Листинг 5

```

<%
String name1=request.getParameter("nm");
String text1=request.getParameter("text");
%>

```

Далее на вкладке **Server Behaviors** панели «**Приложение**» можно выбрать из выпадающего меню пункт **Prepared (подготовлено - Insert, Update, Delete)**, в окне создания запроса сконструировать запрос **INSERT** к базе данных. В результате DreamWeaver вставит JSP-код в страницу, как показано на рисунке 12.



a)

```

14  if(test1 != null){ Prepared1__text1 = (String)test1;}
15
16  -->
17  <%
18  Driver DriverPrepared1 = (Driver)Class.forName(MM_gbook_DRIVER).newInstance();
19  Connection ConnPrepared1 = DriverManager.getConnection(MM_gbook_STRING,MM_gbook_USERNAME,MM_gbook_PASSWORD);
20  PreparedStatement Prepared1 = ConnPrepared1.prepareStatement("INSERT INTO DBA.gbookms (name, texmes) VALUES ('"+name1+"', '"+text1+"' ) "
21  );
22  Prepared1.executeUpdate();
23  <%

```

b)

Рисунок 12 – JSP-код для добавления записей из гостевой книги в базу данных, созданный системой DreamWeaver

На рисунках 13 и 14 показана работа созданного приложения:

Программное обеспечение и оборудование для выполнения работы

Работа выполняется на современном персональном компьютере.

Необходимое программное обеспечение:

- Навигатор Web-страниц (браузер)
- Система Dreamweaver
- Система PowerBuilder (для создания базы данных)
- Сервер баз данных Adaptive Server Anywhere компании Sybase
- Web-сервер Jaguar CTS
- Web-сайт в виде набора HTML-страниц и файлов JavaScript.

Порядок выполнения работы

- 1) Ознакомиться с теоретической справкой.
- 2) Ознакомиться с примером Web-приложения «Гостевая книга», которая взаимодействует с базой данных для просмотра и добавления сообщений посетителей.
- 3) Используя технологию Java Server Pages (JSP) создать web-приложение для работы с базой данных в среде DreamWeaver. Для реализации клиентской части использовать язык Java Script. Для построения внешнего вида страниц приложения - каскадные таблицы стилей (CSS).

Список использованных источников

1. Java Server Pages [Электронный ресурс] / Codenet. - Режим доступа: <http://www.codenet.ru>. - Загл. с экрана.
2. Компьютерный форум OSzone.net [Электронный ресурс] / OSzone.net. - Режим доступа: <http://forum.oszone.net>. - [2001 – 2009]. - Загл. с экрана.
3. Базы данных/ Хомоненко А.Д. [и др.]; под ред. Хомоненко А.Д. - 4-е изд., доб. и перераб. - СПб.: КОРОНА принт, 2004. - 736 с.
4. Холл М., Браун Л. Программирование для Web. Библиотека профессионала. – М.: Вильямс, 2002.