

**Министерство образования и науки Российской Федерации
Государственное бюджетное образовательное учреждение высшего профессионального
образования
«Томский государственный университет систем управления и радиоэлектроники»**

УТВЕРЖДАЮ

Зав.кафедрой ЭС

_____ Н.Е.Родионов
" ____ " _____ 2012 г.

Вводится в действие с " ____ " _____ 20 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ РАБОТ**

по дисциплине

Веб-программирование

Составлена кафедрой

Электронных систем

Для студентов, обучающихся

по направлениям подготовки 222000.62 «Инноватика»

221000.62 «Мехатроника и робототехника»

221400.62 «Управление качеством»

Форма обучения

очная

Составитель вед.программист центра
Международной IT-подготовки ТУСУР

Долматов А.В.

" 20 " сентября 2012 г

Томск 2012 г.

Введение

Лабораторные работы в рамках курса «Web-программирование» предполагает, что студенты будут иметь возможность на практике использовать полный базовый набор технологий, позволяющий создать полнофункциональное web-приложение.

Курс предполагает создание студентами прототипа личного сайта (либо сайта по интересующей студента тематике), технически пригодного к реальному размещению в сети интернет. Предполагается, что полученный в ходе прохождения курса опыт даст студенту возможность в дальнейшем самостоятельно развить полученные знания до уровня, необходимого для практической работы в роли начинающего web-программиста

Общие требования

Лабораторные работы выполняются студентами очной формы обучения индивидуально под контролем со стороны преподавателя. Все консультации осуществляются преподавателем. Число студентов, одновременно присутствующих на занятии не должно превышать 12 человек. Если в списочном составе группы студентов больше 12, то группа должна быть разделена на подгруппы численностью от 6 до 12 человек в каждой.

Для выполнения лабораторных работ целесообразно в учебном расписании выделять 4 академических часа подряд, без больших перерывов. Расписание также должно предусматривать раздельное проведение занятий у подгрупп, если группа была разделена.

Перед началом занятий студенты должны изучить инструкцию по охране труда. Преподаватель должен убедиться в знании инструкции, задавая студенту вопросы по ее содержанию, после чего сделать соответствующую запись в журнале охраны труда.

Во время проведения лабораторных занятий в аудитории (лаборатории) студентам запрещается передавать друг другу файлы и другие материалы, являющиеся результатом выполнения заданий.

При выполнении работ студенты не должны использовать инструментальные средства для html-форматирования – все html-документы должны редактироваться и форматироваться вручную.

Студент имеет право:

- Выходить из аудитории (лаборатории) не спрашивая разрешения у преподавателя.
- Самостоятельно распределять аудиторное время, определяя необходимость перерыва или непрерывной работы.
- Просить консультации у преподавателя, если он в текущий момент не распределяет задания, не принимает выполненные работы и не консультирует другого студента.

Преподаватель, давая консультацию студенту, указывает раздел технической документации или методической литературы, в которой имеется ответ на вопрос студента. Если необходимые сведения в документации и литературе отсутствуют, то преподаватель должен дать устные пояснения или продемонстрировать практические действия, приводящие к требуемому результату, с последующей отменой для повторения студентом.

Самостоятельная работа студентов над лабораторными заданиями осуществляется в той же аудитории (лаборатории), где проводятся лабораторные занятия. Преподаватель должен согласовать со студентами расписание самостоятельной работы - не менее 2 астрономических часов в неделю. В указанное время по учебному расписанию студентов и в аудитории (лаборатории) не должны проводиться другие занятия. Преподаватель должен обеспечить доступ студентов в аудиторию (лабораторию) в указанные часы. Необходимость самостоятельной работы определяет студент.

Консультации, выдача лабораторных заданий и прием результатов выполнения осуществляется только во время аудиторных занятий. Задания выполняются последовательно. Правильное выполнение некоторых заданий возможно только, если студент корректно выполнил предыдущие задания. Поэтому приступать к следующему заданию студент может, только сдав преподавателю результат выполнения предыдущего.

Техническое обеспечение практических работ

Для выполнения лабораторных работ студенту предоставляется индивидуальное рабочее место, в состав которого входят:

- стул;
- стол;
- персональный компьютер с операционной системой MS Windows;
- файл-менеджер с возможностью редактирования файлов (Far Manager или подобный);
- web-браузер, желательно – Firefox или Google Chrome;
- web-сервер Apache с установленным PHP;
- сервер баз данных MySQL;
- доступ в интернет;
- для последнего задания крайне желательным является наличие реального учебного web-сервера в сети интернет с возможностью развёртывания на нём разработанных студентами web-приложений.

Размещение и освещенность рабочих мест в учебной аудитории (лаборатории) должно удовлетворять действующим требованиям СанПиН.

Прием результатов выполнения лабораторных работ

Результаты выполнения лабораторных работ демонстрируются преподавателю. Во время приема выполненной работы преподаватель вправе:

- Требовать у студента демонстрации выполнения программы, предусмотренной заданием.
- Просматривать исходный код программы, составленной студентом.
- Самостоятельно производить манипуляции с интерфейсом программы, не изменяя программы, составленной студентом.
- Требовать у студента пояснений, относящихся к исходному коду и способам реализации программы.

Задание считается выполненным, и принимается преподавателем только в случае, если реализована вся функциональность, предусмотренная заданием. Если какие то функции, предусмотренные заданием, не работают, или работают неверно, то результат выполнения подлежит доработке. Студент должен работать над кодом программы максимально самостоятельно.

До конца семестра студент должен сдать результаты выполнения всех лабораторных работ, предусмотренным настоящими указаниями. В противном случае студенты к сдаче экзамена (зачета) не допускаются.

Задания для лабораторных работ

1. Построение простейшего html-документа.

Задача: построить в текстовом редакторе и продемонстрировать в web-браузере простейший html-документ, содержащий заголовок окна браузера и произвольным образом форматированный текст с заголовком.

Принимается просмотром страницы в браузере и просмотром исходного кода.

2. Форматирование и оформление текста средствами html, вставка изображений.

Задача: в ранее созданном html-документе разбить текст на абзацы, выделить отдельные фрагменты цветом, курсивом, жирным шрифтом, поместить в текст нумерованный список. Поместить в текст изображение, обтекаемое текстом.

Принимается просмотром страницы в браузере и просмотром исходного кода.

3. Компоновка страницы средствами html.

Задача: перекомпоновать текст в ранее созданном html-документе в 2-3 колонки с различным выравниванием (вправо, влево, по середине, по

ширине), обеспечить отступ между колонками в 2 сантиметра. Создать второй html-документ и установить в их тексте гиперссылки друг на друга.

Создать заголовочный блок на ранее созданной html-странице с картинкой и общим заголовком web-приложения, которое будет разрабатываться далее.

Принимается просмотром страницы в браузере и просмотром исходного кода.

4. Оформление html средствами css

Задача: в ранее созданном html-документе с помощью стилей, организованных в css-документ, добавить графическую «подложку», установить общее начертание шрифта для всего документа, оформить цветом и размером заголовки, заменить «буллеты» у нумерованного списка.

Принимается просмотром страницы в браузере и просмотром исходного кода.

5. Настройка PHP и простейшая PHP-программа, методика разработки

Задача: найти и просмотреть конфигурирующие файлы web-сервера и PHP, разместить и открыть ранее созданные html-документы через web-сервер, создать простейший PHP-скрипт, вывести информацию о PHP и «Hello, world!» средствами PHP.

Принимается просмотром страницы в браузере, просмотром исходного кода, просмотром последовательности действий студента в файловом менеджере.

6. Встраивание PHP-скрипта в html-документ, ввод-вывод.

Задача: перестроить ранее созданный html-документ в php-скрипт, внедрить в него простейший вывод – фамилия студента, сегодняшняя дата, параметры адресной строки web-браузера.

Принимается просмотром страницы в браузере, просмотром исходного кода.

7. Применение циклов, условных операторов.

Задача: в зависимости от одного из параметров адресной строки вывести на странице либо то, что сделано в предыдущих лабораторных, либо выведенный в табличном виде календарь на текущий месяц.

Принимается просмотром страницы в браузере, просмотром исходного кода.

8. Применение функций, обработка массивов.

Задача: оформить ранее созданный скрипт вывода календаря в функцию, с двумя параметрами – месяц и год. Интегрировать функцию в ранее созданную страницу. Организовать приёма параметров функции из

параметров адресной строки. Добавить ссылки для перехода к показу следующего и предыдущего месяца.

Рекомендация: для вычисления смещения первого числа месяца по дням недели использовать отправную дату и таблицу количества дней в месяцах.

Принимается просмотром страницы в браузере, просмотром исходного кода.

9. Форматированный вывод массивов, чтение из файла.

Задача: прочитав текст из текстового файла, вывести его в составе страницы. Первую строку, прочитанную из файла, оформить, как заголовок. Последнюю – как подпись под текстом.

Рекомендация: использовать чтение файла целиком.

Принимается просмотром страницы в браузере, просмотром исходного кода.

10. Программирование скрипта показа новостей

Задача: создать текстовый файл, содержащий несколько новостей (каждая новость должна содержать заголовок, дату, краткое содержание, текст, подпись или источник). Организовать с помощью PHP-скрипта чтение файла и вывод – в зависимости от параметров адресной строки – либо полного списка заголовков новостей с датой и кратким содержанием, служащим индексом (то есть, заголовок каждой новости должен являться ссылкой на показ отдельной новости), - либо отдельной новости в полном составе.

Рекомендация: использовать чтение файла целиком и построчный его разбор на новости, одну новость от другой отделять пустой строкой.

Принимается просмотром страницы в браузере, просмотром исходного кода.

11. Построение формы и приём её результатов.

Задача: построить форму, содержащую набор полей, соответствующий набору полей новости из предыдущей работы. Создать скрипт, принимающий поля формы и записывающий их упорядоченно в ранее созданный файл новостей с соблюдением его структуры (по окончании работы скрипта обработки должен выводиться список новостей). Реорганизовать вывод новостей так, чтобы позже записанная новость показывалась вверху списка.

Принимается просмотром страницы в браузере, просмотром исходного кода.

12. Программирование скрипта авторизации пользователя.

Задача: создать текстовый файл, содержащий список пользователей системы (для пользователя указывается логин, пароль, полное имя, год

рождения). Создать форму авторизации пользователя с полями ввода логина и пароля. Создать обработчик формы, проверяющий, есть ли в списке пользователь с введёнными реквизитами, и, если есть, устанавливающим соответствующий флаг в сессии и возвращающий управление на страницу с формой, в противном случае – выводящий сообщение об ошибке. Состояние авторизованности должно отражаться в интерфейсе: в состоянии авторизованности вместо формы авторизации должны показываться имя пользователя и кнопка «Выйти» (по нажатию которой – и только по нажатию которой - происходит выход из состояния авторизованности).

Принимается просмотром страницы в браузере, просмотром исходного кода.

13. Построение структуры web-приложения, разделение прав пользователей

Задача: организовать все ранее созданные в ходе лабораторных работ скрипты в простейшее web-приложение, имеющее следующие страницы: «Главная» (с произвольным текстом, читаемым из файла), «Информация» (с первоначальными учебными элементами оформления), «Новости» (со всей функциональностью новостей), «Календарь» (с функциональностью календаря). Все страницы должны иметь единую структуру и единое визуальное оформление. Меню и форма авторизации должны быть на всех страницах web-приложения. Функциональность каждой страницы web-приложения должна располагаться в отдельном PHP-модуле. Постоянная часть страниц должна находиться только в одном PHP-модуле. Все используемые изображения должны быть сложены в отдельном каталоге.

Принимается просмотром страницы в браузере, просмотром исходного кода, структуры файлов приложения в файловом менеджере.

14. Построение структуры простейшей базы данных и простейшего запроса выборки.

Задача: создать базу данных MySQL, содержащую данные о пользователях системы (минимальный набор полей: идентификатор, логин, пароль, полное имя, год рождения, роль в системе), их сообщениях друг другу (минимальный набор полей: идентификатор, время отправления, отправитель, получатель, текст), новостях (минимальный набор полей: идентификатор, заголовок, дата, краткое содержание, текст, автор). Наполнить тестовыми данными таблицу пользователей (не менее пяти записей). Продемонстрировать в консоли вывод самостоятельно построенный запроса на выборку данных из таблицы пользователей с условием выборки.

Рекомендация: поля отправителя и получателя в таблице сообщений, а также поле автора в таблице новостей, должны быть численными и являться ссылками на поле идентификатора в таблице пользователя.

Принимается просмотром результатом действий студента в консоли базы данных.

15. Установление связи PHP-скрипта с базой данных, вывод данных.

Задача: создать в ранее разработанном web-приложении страницы «Список пользователей», организовать подключение скрипта к базе данных, созданной в предыдущей работе, средствами PHP сделать выборку списка пользователей и отобразить её на странице в табличном виде.

Принимается просмотром страницы в браузере, просмотром исходного кода.

16. Переработка скрипта авторизации на использование базы данных.

Задача: переработать ранее созданный скрипт авторизации пользователя с использования в качестве источника данных файла на использование базы данных.

Рекомендация: в параметрах сессии должны сохраняться идентификатор, имя и роль авторизованного пользователя.

Принимается просмотром страницы в браузере, просмотром исходного кода.

17. Программирование скрипта построения запроса на внесение данных.

Задача: в ранее разработанном web-приложении создать страницу «Сообщения», доступную только авторизованным пользователям. Страница должна содержать список сообщений, адресованных текущему пользователю и форму отправки сообщений, содержащую выпадающий список всех пользователей и поле ввода текста сообщения. По нажатию кнопки «Отправить», сообщение должно корректно вноситься в таблицу базы данных, после чего должна снова выводиться страница сообщений. Для других пользователей чужие сообщения должны быть недоступны.

Принимается просмотром страницы в браузере, просмотром исходного кода.

18. Построение запроса на внесение данных базу, программирование скрипта регистрации пользователя

Задача: в ранее разработанном web-приложении создать страницу регистрации пользователя, доступную только неавторизованному пользователю. Страница должна позволять пользователю внести себя в список пользователей (поле «Роль» не показывается). В случае, если пользователь пытается зарегистрироваться под логином, который уже есть в базе, должно выдаваться сообщение об ошибке. Адаптировать данную страницу для использования пользователем с правами администратора для редактирования свойств пользователя (поле «Роль» показывается в виде выпадающего списка с элементами «Администратор» и «Пользователь»).

Рекомендация: доступ для администратора в редактирование пользователей должен осуществляться через страницу «Пользователи».

Принимается просмотром страницы в браузере, просмотром исходного кода.

19. Переработка скрипта показа и редактирования новостей

Задача: написать скрипт, загружающий в таблицу новостей ранее введённые в файл новости. Создать в странице «Новости» функциональность для пользователя с правами администратора для редактирования, сохранения, создания новых и удаления новостей.

Рекомендация: доступ к редактированию следует делать в рамках уже имеющейся страницы «Новости» - функциональность редактирования должна замещать показ новости. После сохранения или удаления, пользователь должен возвращаться к списку новостей. Для создания новостей следует использовать передачу в страницу редактирования заведомо несуществующего параметра идентификатора новости – например, -1.

Принимается просмотром страницы в браузере, просмотром исходного кода.

20. Интеграция простейшего скрипта на javascript в html-документ.

Задача: ввести в страницу редактирования новости выдачу сообщения «Вы вошли, как администратор, поэтому можете редактировать новость». Ввести в режим удаления новости запрос подтверждения «Вы действительно хотите удалить эту новость?» - с прекращением операции в случае отрицательного ответа.

Принимается просмотром страницы в браузере, просмотром исходного кода.

21. Программирование скрипта показа сообщений об ошибках.

Задача: реконструировать выдачу ошибки при регистрации пользователя (о том, что такой пользователь в базе уже есть) с выдачи в теле страницы на выдачу во всплывающем окне сообщения. Сообщение должно выводиться после отрисовки страницы в браузере.

Принимается контролем функциональности страницы в браузере.

22. Программирование скрипта проверки заполненности полей формы.

Задача: дополнить функциональность регистрации пользователя вводом контрольной копии пароля, контролем заполненности полей, контролем совпадения пароля с его контрольной копией – с выдачей соответствующего сообщения и остановкой операции в случае, если заполнены не все необходимые поля, или пароль не совпадает с его контрольной копией.

Принимается контролем функциональности страницы в браузере и просмотром исходного кода.

23. Интеграция JQuery и те же скрипты, переписанные с использованием JQuery

Задача: подключить к web-приложению JQuery и переписать функциональность предыдущего задания с его помощью.

Принимается контролем функциональности страницы в браузере и просмотром исходного кода.

24. Программирование динамического меню с помощью JQuery

Задача: добавить к меню приложения второй уровень – в пункте «Новости» добавить заголовки последних пяти новостей, в пункте «Информация» добавляются подпункты «О себе» и «О проекте» (ведущие на страницы, содержащие надпись «Страница в разработке»). Второй уровень должен выпадать при наведении курсора мыши на заголовок соответствующего пункта верхнего уровня.

Рекомендация: использовать для выпадающих панелей тэг div со стилями display:none и position:absolute. Кроссбраузерности добиваться не требуется.

Принимается контролем функциональности страницы в браузере и просмотром исходного кода.

25. Общее оформление web-приложения.

Задача: оформить разработанное web-приложение с помощью стилей до состояния минимальной пристойности внешнего вида.

Рекомендация: обращать внимание на наличие и стиль заголовков, выравнивание текста, отступы, рамки, единообразие вывода текста.

Принимается контролем внешнего вида страницы в браузере.

26. Развёртывание сайта на хостинге.

Задача: перенести разработанное web-приложение на реальный хостинг с назначением ему доменного имени. В случае невозможности – просто перенос разработанного приложения в подкаталог учебного web-сервера без индивидуального доменного имени с полным сохранением функциональности.

Принимается контролем функциональности в браузере.

Библиографический список

Соответствует списку литературы из рабочей программы.