

Технологии программирования для сети Internet / Технологии разработки веб-сервисов

Степан Овчинников, Алексей Шкарупа, Анатолий Ерофеев, ИНТЕРВОЛГА

О курсе

Продолжительность: 2 семестра.

Обучение не предполагает глубокого разбора на лекциях синтаксиса конкретных языков и технологий. Учим принципам, даем задание, помогаем разобраться.

На чем базируется курс:

- умение писать код на алгоритмическом языке программирования
 - Примеры задач: сортировка, поиск, вывод массива, поиск наибольшего квадрата с нулями, поиск самого длинного предложения в тексте
 - Уверенный процедурный стиль, Основы ООП
- основы sql из курса БД
- понимание работы сетевого взаимодействия: ip, порт, протокол tcp
- способность самостоятельно изучать документацию по технологиям.

Цель курса:

- На выходе — уверенное понимание принципов работы веб-систем, способность разработать веб-приложение (1 семестр)
- Понимание современных технологий веб-разработки и механизмов интеграции в веб-системах, способность строить интегрированные системы (2 семестр)

Подход к построению курса.

Учим всем веб-технологиям параллельно: клиент, сервер, протокол, серверные и клиентские языки. По мере движения вперед усложняются задачи. Первый семестр — решаем все типовые задачи на чистых технологиях, второй — разбираем примеры реальных задач на фреймворках.

Стартовый уровень студентов:

- C++
- Математика
- Консольные приложения

Одновременно учат

- БД
- ООП (в конце)
- qt

1 семестр (2019/2020)

Кодовое название семестра — “Понюхай порох разработки на чистом linux, apache, php, mysql без фреймворка”

Номер лекции	Тема лекции 12 - СО 5 - АШ	Задание на лабораторные работы 5 - СО 3 - АШ/АЕ
--------------	----------------------------------	---

<p>1</p>	<p>Особенности интернета как среды работы приложений. Чем веб-разработка отличается от всего остального мира ПО. Обзорная лекция по курсу Технические особенности веб-разработки:</p> <ul style="list-style-type: none"> - Клиент и сервер - Http и stateless-приложения - Сессии и cookies - Доступ к http-заголовкам, cookies, сессиям из программного кода - Необходимость разделение логики разного типа: <ul style="list-style-type: none"> - html, - серверный код, - клиентский код, - код запросов СУБД - Сборка интерфейса на сниппетах кода (введение в bootstrap) <p>Вспомогательные презентации: Cookbook Веб для бизнеса Особенности веба и архитектура веб-систем Заготовка презентации "как сделать VK на PHP" Древняя презентация про bootstrap</p> <p>Лекция 1: Введение в интернет-проекты, технические особенности веб-разработки, HTML, HTTP и bootstrap.</p> <p>СО</p>	<p>Перед циклом лабораторных работ студенты разбиваются на группы по 2 человека. Каждая группа получает "тему":</p> <ol style="list-style-type: none"> 1. Библиотека 2. Больница 3. Автобаза 4. Магазин 5. Шиномонтаж <p>...</p> <p>И делает все задания с учетом специфики.</p> <p>ЛР 1 Основы веб-разработки HTML и HTTP Сборка простого интерфейса на bootstrap Делаем руками http-запрос через терминал и изучаем заголовки</p> <p>10 баллов АШ/АЕ</p>
<p>2</p>	<p>Введение в разработку на PHP</p> <ul style="list-style-type: none"> ● PHP-C-подобный язык со слабой типизацией и низким порогом входа ● Синтаксис языка PHP ● Типы данных, переменные ● Глобальные переменные ● Выражения ● Операторы ● Побитовые операторы ● Функции ● Классы ● Отображение ошибок ● Авторизация на сайте <p>СО</p>	
<p>3</p>	<p>Основы SQL и СУБД MySQL https://docs.google.com/presentation/d/12yICzXxSnLvXvVfYg4CpfNwyarJi7ED4cDX6RaYqK_I/edit#slide=id.p (1 лекция отсюда)</p> <p>СО</p>	<p>ЛР 2 Тема: Stateless HTTP-протокол и его реальное применение. Механизм авторизации: сессии, куки, предоставление доступа.</p>
<p>4</p>	<p>Основы разработки на PHP под веб-сервер</p> <p>Обработка формы обратной связи с</p>	<p>Задание: Реализовать на чистом PHP и HTML/css механизм авторизации и простого личного кабинета.</p>

	<p>сохранением данных и валидацией Работа с загружаемыми файлами в PHP Регулярные выражения Безопасность файловой системы Безопасность в веб-разработке вообще Основные директивы php.ini</p> <p>Презентация Безопасная веб-разработка</p> <p>СО</p>	<p>Написать простой скрипт на php, который будет авторизовывать только пользователей из php-массива (логин => пароль) Объявить ассоциативный массив на php вида (логин => пароль). Вывести форму авторизации (Логин + пароль + кнопка ""Войти"").</p> <p>Написать скрипт авторизации пользователя. Выводить ошибки авторизации (""Логина нет в системе"", ""Неверный пароль"").</p> <p>Если пользователь авторизован -- вывести время входа и кнопку/ссылку ""Выйти"", которая завершает сессию.</p> <p>Контрольные вопросы: Какой тип запроса к веб-серверу следует использовать для отправки данных авторизации? Что такое хэш-функция? Как защищаться от SQL-инъекций?</p> <p>10 баллов СО</p>
5	<p>Основные "сопутствующие веб-технологии"</p> <p>XML/csv/json AJAX htaccess robots.txt Sitemap.xml</p> <p>СО</p>	<p>ЛР 3 Тема: особенности разработки на PHP</p> <p>Ассоциативные массивы, регулярные выражения, работа с HTML</p> <p>10 баллов СО</p>
6	<p>Основы создания веб-серверов на Unix</p> <p>Работа по ftp Перезапуск Unix-сервера Предназначение LAMP</p> <p>Работа по SSH/Putty Просмотр списка процессов, остановка процесса, перезапуск apache Работа с файловой системой Установка VMBitrix CRON Установка и настройка LAMP</p> <p>Установка Битрикс веб-окружение</p> <p>АШ</p>	
7	<p>Проектирование веб-систем. ТЗ, БД, интерфейсы, сценарии</p> <p>СО</p>	<p>ЛР 4 Реализуем интерфейс управления данными. CR (без UD) на одной таблице Требования к хорошему решению см. в пункте 5б отсюда</p>
8	<p>Взаимодействие с базой данных на PHP</p> <p>СО</p>	<p>https://www.intervolga.ru/vacancy/developer/junior.php</p>

		10 баллов СО
9	Методы интеграции веб-приложений с внешними системами. СО	ЛР 5 Реализуем интерфейс управления данными. CRUD 2 таблиц полностью
10	Основы фронтенда Что такое JavaScript? 3 способа подключения скрипта Операторы Функции Jquery на уровне "селекторы" и "обработчики событий" Отладка JS с помощью консоли браузера Создание форм на html Синтаксис и селекторы CSS Интеграция с GA/Я.Метрика	Добавляем хранение файла как свойства объекта. Работа с загружаемыми файлами в PHP 10 баллов СО
11	Продолжение фронтенда: Bootstrap, Работа с плагинами jquery и AJAX [ES6 и React/Vue - если хватит времени] Презентация про ajax в битриксе АШ	ЛР 6 Собственная регистрация, поиск, авторизация через таблицу users, внешняя авторизация. Задание: добавляем к CRUD 2 таблиц поиск по обеим сущностям, регистрацию вручную и через VK/FB, как внешний источник авторизации и интеграцию с любым внешним сервисом 1. любым сервисом с tech.yandex.ru a. рекомендую Спеллчекер, Предиктор, Карты, Алиса 2. любым государственным сервисом с открытыми данными a. рекомендую ФИАС b. https://data.gov.ru/ 3. Dadata.ru 4. Сервисы с открытым API a. bitrix24 — получить список задач по номеру пользователя b. amoCRM — получить список дел на сегодня по номеру пользователя c. ...
12	Реализация пошаговых механизмов в вебе (установщики, конвертеры и прочее) АШ	15 баллов АШ/АЕ
13	Продвинутая разработка на PHP Основы ООП, написание примера на каждого "слона" (ИНП) Применение паттернов (Singleton, ACL, Фабрика) Обработка ошибок и исключений Интерфейсы и абстрактные классы Применение пространства имен работа с модулями PHP (cURL, DOM, GD)	ЛР 7 Добавляем к пред. Разработке еще 5 таблиц, придумываем 6 отчетов, реализуем систему отчетов. Добавляем логирование отчетов пользователя 15 баллов СО

	АШ	
14	Методы отладки в веб-разработке на PHP под Apache Xdebug, xhprof СО	
15	Методы ускорения работы медленных запросов в СУБД СО	ЛР 8 Переделываем результат 6 ЛР (CRUD 2 таблиц) на ajax и работу в режиме single page application. 20 баллов АШ/АЕ
16	Разработка UNIT-тестов Разработка SELENIUM-тестов Введение в Continuous Integration АШ	
17	Обзорная завершающая лекция Задачи современной веб-разработки и особенности их решения Презентация про git Презентация про автоматическое тестирование Методы оценки проектов СО	

Задания “со звездочкой” (можно сдать 6,7,8 всех лаб -- пачкой)

1. CRUD 2 таблиц с правами и авторизацией с таблицей БД, логированием и транзакциями.
2. Добавить внешнюю интеграцию с одним из сервисов Яндекса (tech.yandex.ru).
3. Добавляем к любому реальному проекту на LAMP вертикальный шардинг на базе кластера на MySQL. Ввод и управление данными на 1 ноде, отчетность и просмотр статистики на 2 ноде.
4. Мониторинг и нагрузочное тестирование любого реального проекта на LAMP.
5. Реализация в “любимом” фреймворке одного из “мастеров” Битрикса:
 - a. Резервное копирование
 - b. Антивирус
 - c.

2 семестр (2019/2020)

(34 ч = 17 лекций, 1 лекция в неделю),

Кодовое название семестра — “А теперь все то же самое промышленными способами”.

Последовательно разберем все прикладные вопросы промышленной веб-разработки и изучим технологии

Номер лекции	Тема лекции	Примерное задание на лабораторные работы Часть -- Анатолий
1	Реальные задачи веб-разработки и роль фреймворков (обзор): <ul style="list-style-type: none"> - CRUD - Веб-формы - Корзина - Древовидные комментарии - Гибкость или 	ЛР 2 семестра заключаются в последовательном прохождении Академии 1С-Битрикс ЛР 1 1. Установка платформы

	<p>производительность при управлении данными</p> <ul style="list-style-type: none"> - Настраиваемое отображение - Кеширование - Виджеты - Агенты и регламентные задачи - Многосайтовость и кроссдоменная авторизация - Локализация - Личные кабинеты пользователя, интегрированные с внешними источниками данных - Геозависимость <p>СО</p>	<p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5872&LESSON_PATH=5862.5872</p> <p>2. Структура Bitrix Framework</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5873&LESSON_PATH=5862.5873</p> <p>3. Интеграция HTML шаблона</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5874&LESSON_PATH=5862.5874</p> <p>4.1 Компоненты. Часть 1</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5875&LESSON_PATH=5862.5875</p> <p>4.2 Компоненты. Часть 2</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=6742&LESSON_PATH=5862.6742</p> <p>5. Информационные блоки - обзорно, настройки.</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5876&LESSON_PATH=5862.5876</p>
2	<p>Обзор-сравнение современных фреймворков:</p> <ul style="list-style-type: none"> • PHP <ul style="list-style-type: none"> ○ Yii 1, 2 ○ Laravel ○ Symfony ○ CodeIgniter ○ CakePHP ○ Bitrix ○ Zend • Python <ul style="list-style-type: none"> ○ Django и Flask <p>АШ</p>	<p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5876&LESSON_PATH=5862.5876</p>
3	<p>Архитектура Bitrix Framework</p> <p>АШ</p> <p>Презентация про бекапы</p>	<p>ЛП 2</p> <p>6. Вывод динамической информации</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5877&LESSON_PATH=5862.5877</p>
4	<p>CRUD и методы решения задачи</p> <p>Настраиваемое отображение структурированных динамических данных с логикой</p> <p>Компоненты и их устройство</p> <p>СО</p>	<p>7. Кеширование. Основы работы.</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5879&LESSON_PATH=5862.5879</p> <p>8. Комплексные компоненты</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5879&LESSON_PATH=5862.5879</p> <p>9. ЧПУ, "эрмитаж", постраничная навигация</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5880&LESSON_PATH=5862.5880</p> <p>10. Поиск</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=5881&LESSON_PATH=5862.5881</p> <p>11. Полезные возможности</p> <p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&LESSON_ID=6743&LESSON_PATH=5862.6743</p>
5	<p>Гибкость и производительность при</p>	<p>ЛП 3</p>

	<p>организацию хранения данных Презентация про инфоблоки</p> <p>СО</p>	<p>Проектирование веб-приложений на примере своей курсовой: Структура БД, сценарии, представления, сборка верстки на bootstrap, описание внешней интеграции</p>
6	<p>Кеширование Комплексные компоненты</p> <p>АШ</p>	
7	<p>“Админка” и методы редактирования контента в веб-разработке</p>	<p>ЛР 4 2 часть</p>
8	<p>Поиск и постраничная навигация События, агенты (upd 31.03.2020)</p>	<p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5894&LESSON_PATH=5863.5894 1.1 Расширение возможностей типовых компонентов http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5884&LESSON_PATH=5863.5884 1.2 Получение связанных элементов http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=7049&LESSON_PATH=5863.7049 2. Технологии: параметры модулей, почтовая системы, агенты, журнал событий http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5885&LESSON_PATH=5863.5885 3. События http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5886&LESSON_PATH=5863.5886 4. Отложенные функции http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5887&LESSON_PATH=5863.5887 5.1 Создание собственных компонентов http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5888&LESSON_PATH=5863.5888 5.2 Настройки кеширования - базовые понятия http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5890&LESSON_PATH=5863.5890</p> <p>Поиск через Sphinx, с индексатором и поиском по каталогу</p>
9	<p>Корзина заказа Гаджеты Отложенные функции, поиск (08.04.2020)</p>	<p>ЛР 5 6. Создание собственного комплексного компонента http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5890&LESSON_PATH=5863.5890 7. Управление доступом и безопасность</p>
10	<p>Веб-формы Древовидные комментарии (22.04.2020)</p>	<p>http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5889&LESSON_PATH=5863.5889 8. Создание "гаджетов" http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5891&LESSON_PATH=5863.5891</p> <p>Работа с деревьями на Bitrix API</p>
11	<p>Агенты и регламентные задачи Многосайтовость и кроссдоменная авторизация Локализация</p>	<p>ЛР 6 9. Создание "мастеров" http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5892&LESSON_PATH=5863.5892</p>

	Интернет-магазин, часть 1 (товары, цены) (29.04.2020)	10. Многосайтовость и многоязычность http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=6451&LESSON_PATH=5863.6451 Итог. Памятка разработчику
12	Работа с почтой в веб-проектах Интернет-магазин, часть 2 (корзина, покупатели, заказы) (06.05.2020)	http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=66&LESSON_ID=5893&LESSON_PATH=5863.5893 Дополнительно работа с почтой усиленно.
13	SEO и интернет-маркетинг для разработчика Интернет-магазин, часть 3 (практика)	ЛР 7-8 Обобщающая работа: Делаем веб-проект на “свою тему” на Bitrix framework.
14	Пользовательские поля Экспорт и импорт Интернет-магазин, часть 4 (API)	
15	Производительность проектов на Bitrix Framework Монитор качества Технология Композитный сайт Пользовательские поля Импорт и экспорт	
16	Личные кабинеты Интеграция IT-систем	
17	Личные кабинеты пользователя; интегрированные с внешними источниками данных Веб-сервисы, REST и SOAP	

Лабораторная работа 1. “Основы веб-разработки HTML и HTTP”

Требуемое ПО

- <https://www.apachefriends.org/ru/index.html>
- удобный текстовый редактор Notepad++ или любая IDE с поддержкой PHP, HTML, CSS, JS, SQL
- современный браузер Chrome/Firefox

Суть работы

- Рисуем сетки интерфейсов
- Собираем простой интерфейс на bootstrap
- Делаем руками http-запрос через терминал и изучаем заголовки

Цель работы:

Познакомиться с основными понятиями веб-интерфейсов, языком HTML и протоколом HTTP.

На 1 ЛР студенты разбиваются на группы по 2 человека. Каждая группа получает “тему” - про что будем делать сайт в течение всего семестра. Указана тема и основные сущности, которые на сайте должны быть представлены.

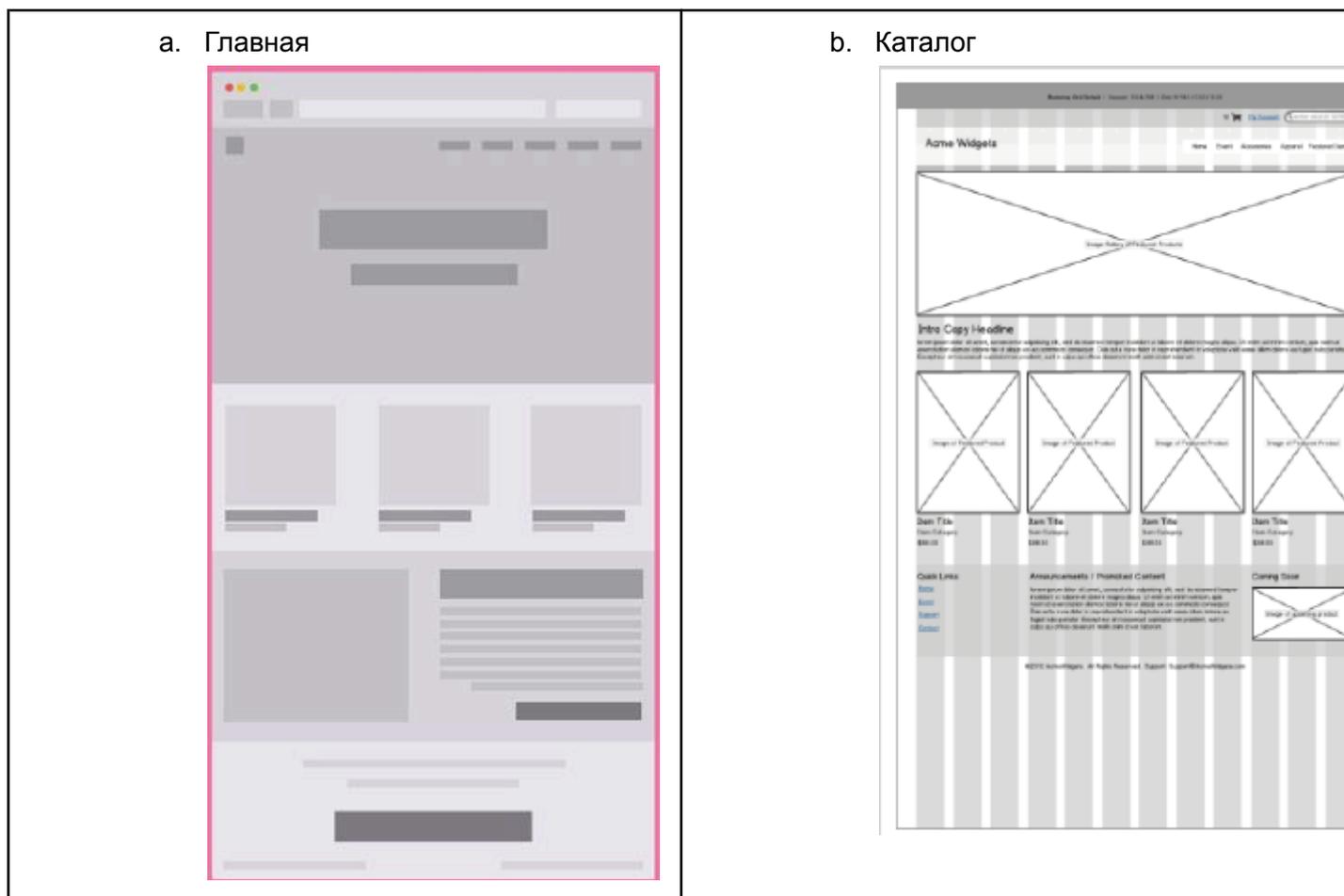
Темы повторяться не должны.

1. Обменный пункт: сотрудники пункта, виды валют, курсы валют, операции обмена.
2. Ювелирный магазин: названия изделий, комитенты (кто сдал изделия на комиссию), журнал сдачи изделий на продажу, журнал покупки изделий.
3. Поликлиника: врачи, пациенты, виды болезней, журнал учета прихода пациентов.
4. Кондитерский магазин: виды конфет, поставщики, торговые точки, журнал поступления и отпуска товара.
5. Автобаза: автомашины, водители, рейсы, журнал выезда машин на рейсы.
6. Парикмахерская: клиенты, прайс услуг, сотрудники, кассовый журнал.
7. Склад: поставщики товара, список товара, получатели товара, кладовщики.
8. Школа: учителя, предметы, ученики, журнал успеваемости.
9. Оплата услуг на дачных участках: виды услуг, список владельцев, сотрудники управления, журнал регистрации оплат.
10. Гостиница: проживающие, сотрудники гостиницы, номера, журнал регистрации проживающих.
11. Книжный магазин: авторы, книги, продавцы, покупатели, регистрация продаж.
12. Ремонтная мастерская: виды работ, исполнители, заказы на ремонт, заказчики.
13. Аптечный киоск: номенклатура лекарств, работники аптеки, покупатели, журнал регистрации продаж.
14. Выставка: стенды, стендисты, экскурсии, посетители.
15. Охранная служба: список постов охраны, список охранников, журнал выхода на дежурство, журнал учета замечаний.
16. Столовая: продукты, блюда, меню, журнал заказов
17. Фото мастерская: заказчики работ, прайс работ, журнал поступления заказов, исполнители.
18. Ветеринарная лечебница: список животных, список болезней, список хозяев, журнал посещений.
19. Сельское хозяйство: список растений, список угодий, список работников, журнал посевной.
20. Холдинг: список регионов, список предприятий, список показателей, журнал учета отчетных данных.
21. Фонды предприятия: список основных средств, список категорий основных средств, список материально ответственных лиц, журнал учета состояния основных средств.
22. Учет расхода материалов в компании: список статей затрат, список сотрудников, журнал учета расхода канцтоваров, список департаментов.
23. Фильмотека: список фильмов, список клиентов, список библиотекарей, журнал выдачи фильмов.
24. Цирк: список категорий артистов, список артистов, журнал выхода артистов на работу, список цирковых площадок.

25. Спортивные заведения: список спортсменов, список видов спорта, список стадионов, журнал учета выступлений спортсменов.
26. Компьютерные занятия: список слушателей курсов, список предметов, список преподавателей, журнал учета успеваемости.
27. Сбор урожая: список видов продукции, список сборщиков, список бригад, журнал учета сбора урожая.
28. Фирма по обслуживанию населения: список заказчиков, список товаров, список разносчиков, журнал заказов.
29. Партийная работа: список членов партии, список мероприятий, журнал учета выхода на мероприятие, список городов
30. Экономическая база данных: список регионов, список показателей, список отраслей, отчетные статистические данные.
31. Журнальные статьи: список тем, список авторов, список названия статей, список журналов.
32. Анализ причин заболеваемости: список больных, список болезней, список районов, журнал учета заболевших.
33. Отдел кадров: список сотрудников, штатное расписание, список отделов, журнал перемещения сотрудников по службе.
34. Делопроизводство: список видов документов, карточка документа, список исполнителей, список департаментов
35. Расчет нагрузки на преподавателя: список преподавателей, список кафедр, предметов, журнал нагрузки.
36. Проектные работы: список проектов, список специалистов, список должностей, журнал учета работ.
37. Учет компьютерного оборудования: список типов оборудования, список материально ответственных лиц, список департаментов, журнал регистрации выдачи оборудования.
38. Прививки детям: список прививок, список детей, список родителей, журнал учета сделанных прививок.
39. Начисление налогов в бюджет: виды налогов, список отраслей, список предприятий, журнал учета поступления налогов.
40. Экспертная система: список оцениваемых объектов, список экспертов, список регионов, журнал учета оценок.
41. Ремонтная мастерская электронного оборудования: список работ, список мастеров, список запасных частей, журнал учета выполненных работ, список поступившего оборудования.
42. Магазин по продаже автомобилей: список фирм производителей, список автомобилей, журнал поступления автомобиля, список водителя пригнавшего машину.
43. Автомобильный гараж: список владельцев, список автомобилей, список сторожей, журнал прихода и ухода автомобилей.
44. Учет криминогенной ситуации в городе: список районов, список типов преступлений, список дежурных, журнал регистрации преступлений.
45. Система здравоохранения: список регионов, список санаториев, список пенсионеров, журнал регистрации выдачи путевок в санатории.
46. Туристические агентства: список туров, список стран, список клиентов, журнал регистрации продаж туров.
47. Продажа билетов на рейсы: список рейсов, прайс билетов, список компаний, журнал продаж билетов.
48. Продажа пиломатериалов: виды пиломатериалов, регионы поставщики, список заказчиков, журнал учета продаж пиломатериалов.
49. Склад металлоконструкций: прайс товара металлоконструкций, список поставщиков, список продавцов, журнал учета продаж.
50. Система поддержки решений: список экспертов, список тем обсуждений, список департаментов, журнал учета предложений.
51. Детский сад: список родителей, список детей, список групп, журнал посещения детского сада.
52. Дом творчества молодежи: список кружков, список руководителей, список детей, журнал регистрации посещения кружков.

Нужно сделать следующие работы:

1. Описать текстом минимум 3 use case (сценария использования) сайта. Например
 - a. Гость может зайти на сайт библиотеки и посмотреть режим работы
 - b. Гость может посмотреть книги в наличии в разделе “каталог”
 - c. Гость может зарегистрироваться на сайте и в личном кабинете “отложить” себе несколько книг
2. Нарисовать (в любой программе, можно в google docs, <https://ninjamock.com/>, draw.io) “сетки дизайна” для минимум 3 страниц будущего сайта. Например



Рисовать можно с любым качеством, главное — суть.

3. Собрать эти 3 страницы на bootstrap <https://getbootstrap.com/>
// Есть русский <https://bootstrap-4.ru/> это альфа-версия, для отстающих и не читающих по-английски

Можно и нужно использовать примеры <https://getbootstrap.com/docs/4.3/examples/>

4. Установить и запустить в XAMPP (это мини-веб-сервер должен быть локально установлен, я писал в требованиях к ПО) <https://www.apachefriends.org/ru/index.html>
 - a. Залить туда верстку этих трех файлов с сопутствующими картинками и стилями, связать простыми ссылками типа
 - i. С главной на страницу каталога и на страницу контактов
 - ii. Со страницы каталога на главную
 - iii. Со страницы контактов на главную
 - b. Открыть localhost в браузере, походить по ссылкам, посмотреть как все это работает
5. С помощью telnet открыть свой свежеделанный сайт из трех страниц с использованием протокола http (https://ru.wikipedia.org/wiki/Telnet#Telnet_%D0%B8_%D0%B4%D1%80%D1%83%D0%B3%D0%B8%D0%

[B5_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B](https://www.shellhacks.com/ru/telnet-send-get-head-http-request/)) примерно как тут описано <https://www.shellhacks.com/ru/telnet-send-get-head-http-request/> (должен быть в операционной системе). Где брать telnet

- a. В win-системах находится в %SystemRoot%\System32\telnet.exe
 - i. Если нет, включается так (права админа нужны)
`dism /online /Enable-Feature /FeatureName:TelnetClient`
- b. В *nix доступно откуда угодно
- c. В mac OS-системах легко ставится

<http://blog.sedicomm.com/2018/01/20/kak-ustanovit-vernut-telnet-na-mac-os-high-sierra-10-13/>

6. С помощью ручного ввода команд получить
 - a. Заголовки каждой страницы через HEAD, проверить статус ответа
 - b. Содержимое каждой страницы через GET, проверить статус ответа
 - c. Заголовок и содержимое несуществующей страницы, например /any_page.html
 - d. Попробовать отправить POST-запрос на главную страницу

Работа сдается в виде ТЕКСТОВОГО ФАЙЛА-протокола:

1. "Тема" (автосервис, например)
2. Описание use case
3. Сетки страниц
4. Верстка HE архивом, а ссылкой на репозиторий
5. Скриншоты как у вас в браузере ваши страницы открываются, скриншоты страниц верстки в протоколе
6. Скриншоты работы в telnet
 - a. Заголовки каждой страницы через HEAD, проверить статус ответа
 - b. Содержимое каждой страницы через GET, проверить статус ответа
 - c. Заголовок и содержимое несуществующей страницы, например /any_page.html
 - d. Попробовать отправить POST-запрос на главную страницу

Лабораторная работа 2. Stateless HTTP-протокол и его реальное применение.

Механизм авторизации: сессии, куки, предоставление доступа, переадресация, простейший запрос к БД.

Задание: Реализовать на чистом PHP и HTML/css механизм авторизации и простого личного кабинета. Написать простой скрипт на php, который будет авторизовывать только пользователей из php-массива (логин => пароль)

Для этого объявить ассоциативный массив на php вида (логин => пароль). Вывести форму авторизации (Логин + пароль + кнопка ""Войти"").

Написать скрипт авторизации пользователя. Выводить ошибки авторизации (""Логина нет в системе"", ""Неверный пароль"").

Если пользователь авторизован -- вывести "секретные" данные из таблицы базы данных и кнопку/ссылку ""Выйти"", которая завершает сессию.

Убедиться что страница с базой данных недоступна по прямой ссылке до авторизации и после завершения сессии.

Ход работы

1. Опираясь на тему и use case первой лабораторной работы, подробно описать use case авторизации
 - a. Внешний вид формы логина

- b. Поля формы логина
 - c. Внешний вид страницы ошибки
 - d. Внешний вид страницы после авторизации.
 - e. Какие данные сайт показывает после авторизации
 - f. Как выглядит и как работает кнопка “выход”
2. Собрать в стиле 1 ЛР страницы
 - a. Неавторизованного пользователя
 - b. Формы авторизации
 - c. Ошибки авторизации
 - d. После авторизации - страница (можно 2-3) личного кабинета
 - e. После выхода
 3. Создать таблицу (*можно 2-3) базы данных для показа “секретных данных” через phpmyadmin. Опирайтесь на требования из темы.
 4. Создать ассоциативный массив логинов-паролей в РНР через текстовый редактор.
 5. Написать на РНР программный код для решения задач
 - a. Обработки кнопки “вход”
 - b. Показа данных в личном кабинете. Данные пусть соответствуют теме.
 - c. Обработки кнопки “выход”
 6. Собрать проект в готовое решение.

Контрольные вопросы:

1. Какой тип запроса к веб-серверу следует использовать для отправки данных авторизации?
 - a. Опишите почему нельзя использовать другие?
 - b. Покажите скриншотом или конкретным кодом в вашем проекте что будет если использовать неправильный метод.
2. Что такое хэш-функция? Для чего она нужна в механизме авторизации.
 - a. Приведите место в код вашего проекта где уместно использование хеш-функции.
3. Какие “уязвимости” в созданном механизме авторизации вы видите? Покажите скриншот, показывающий найденную уязвимость и код, которым вы его закрыли. Подсказки
 - a. Прямая ссылка
 - b. “Мусорные данные в авторизации”
 - c. Хранение недопустимых данных в cookie
4. Как сделать чтобы после авторизации обновление страницы не вызывало запрос браузера вида “требуется повторная отправка формы”?
 - a. Покажите кусок кода, которым вы убрали это сообщение.
5. Чем отличается авторизация на профессионально сделанном сайте от того, что сделали вы?
 - a. Выберите проект по той же теме, по которой у вас лабораторная.

Работа сдается в виде

1. Файлов проекта (верстка+php-код+дамп базы данных), загруженных на любой открытый репозиторий (рекомендуется <https://github.com/> или <https://bitbucket.org/>)
В отчет вставляется ссылка на репозиторий.
2. Скринкаста работы интерфейса (рекомендуется <https://obsproject.com/>)
3. Письменных ответов на контрольные вопросы в отчете

Лабораторная работа 3. Решение прикладных задач на РНР.

Вариант определяется как остаток от деления своего варианта на 10.

Например если ваша тема 33 -- ваш вариант 3

Задачи:

0. Напишите функцию, на вход которой подается текст новости и адрес ссылки на другой сайт.

Нужно вернуть текст, обрезанный многоточием на 180-символе и сделать последние 2 слова текста ссылкой на переданный адрес.

Подумать о проблемах и решить их корректно.

Продемонстрировать на странице сайта, сделанного на 1-2 ЛР

1. На вход вашей функции передается html-текст.
Верните автоматически построенное оглавление со ссылками на заголовки якоря (тег a).
Продемонстрировать на странице сайта, сделанного на 1-2 ЛР.
2. Реализовать функцию, которая автоматически удаляет мат из переданного ей текста.
Список фильтруемых выражений должен быть пополняемый.
Требования: учет падежей и защита от ложных срабатываний типа “употреблять” => “употре#####”
Продемонстрировать на странице сайта, сделанного на 1-2 ЛР
3. Запрограммировать арифметический калькулятор в виде веб-приложения.
Продемонстрировать на странице сайта, сделанного на 1-2 ЛР
4. На вход вашей функции передается массив с данными: (поля данных придумайте исходя из темы ЛР) и отдельным параметром -- число элементов, выводимых на страницу.
Функция выводит корректно работающую “постраничную навигацию” для переданного списка.
Должно работать первая, предыдущая, страницы по номерам, следующая, последняя.
Продемонстрировать на странице сайта, сделанного на 1-2 ЛР
5. Написать код, который автоматически построит sitemap.xml по файлам, физически представленным в document_root папке вашего сайта.
Параметры функции: “входить ли в папки”, “список поддерживаемых типов файлов”
Результат -- файл на диске.
6. Реализовать механизм поиска слов по html-файлам, лежащим в папке /html/ вашего сайта.
Выводить результаты вместе с подсветкой найденного фрагмента.
Продемонстрировать на странице сайта, сделанного на 1-2 ЛР
7. Написать функцию, которая будет автоматически выводить “гlossарий” на основании анализа файлов, /html/ вашего сайта.
Если слово встречается в файлах более N раз, функция добавляет его в glossарий. Клик на слове выдает список ссылок на файлы, где это слово встречается.
8. Реализовать функцию safety_request, делающую безопасными данные POST и GET-запросов.
Делать безопасным JS, SQL, экранировать HTML.
Встроить в сайт, сделанный в 1-2 ЛР.
9. Реализовать простую систему учета просмотров страниц вашего сайта (результат 1-2 ЛР).
Отслеживать источники переходов на сайт, считать уникальных посетителей и просмотры. Поисковых ботов не считать.
Данные складывать в БД или текстовый файл.
Сделать страницу просмотра статистики.

Общие требования:

- Выделять функции, не писать тонну кода в поток
- Разделять логику и представление
- Не использовать рhr-фреймворков

Задачи намеренно сформулированы недостаточно конкретно.

Требуется додумать задачу, сформулировать ее, реализовать и сдать.

Работа сдается в виде протокола (отдельный текстовый файл):

1. Детальная формулировка задачи с описанием “как будем делать”
2. 3 Тестовых примера со скриншотами, макетами, описанием результатов. Верстаем, рисуем
3. РНР-Код, вставленный прямо в протокол
4. Работающее решение в виде папки с файлами.
5. Скринкаст работы вашего решения

Одинаковые решения принимать не буду.

Лабораторная работа 4. Реализуем интерфейс управления данными. CR (без UD) на одной таблице

Задание:

В таблице базы данных хранятся некие объекты (например, страны).

Ваш объект должен следовать из темы ЛР.

Напишите на html, css, js, php интерфейс добавления нового объекта в таблицу и просмотра списка объектов из таблицы.

Структуру таблицы придумайте самостоятельно. Удаление и редактирование делать не нужно.

Требования:

- “Тематический” внешний вид
- Bootstrap использовать можно
- Минимум 5 полей у объекта
- Минимум 3 типа полей
- Обязательно использовать js-валидацию полей
- Код в соответствии с требованиями (см лекцию)

Ход работы

1. Пишем ТЗ “как надо”
 - a. Структура данных
 - b. Все представления
 - c. Все модульные сетки
 - d. Сценарии работы
 - e. Требования к безопасности
2. Пишем код и собираем решение на github, в протокол ссылку на проект
3. Записать скринкаст как все это работает

Работа сдается в виде протокола:

- ТЗ
- Ссылка на github (дамп базы там же)
- Скринкаст

Задание ультрапростое, поэтому буду смотреть качество кода.

Лабораторная работа 5. Реализуем интерфейс управления данными.

CRUD 2 связанных таблиц полностью

Добавляем к 4 ЛР новые функции:

- Удаление объектов
- Редактирование объектов

Добавляем второй элемент данных, связанный с первым

Необходимо чтобы можно было при просмотре одного элемента посмотреть второй и наоборот -- навигация по ссылкам в соответствии со связями. Передавая ID, думайте о sql-инъекциях.

Пример: страны и города, факультет и группы, группы и студенты, покупатели и заказы.

Добавляем хранение файла как свойства объекта. См лекцию “Работа с загружаемыми файлами в PHP”.

Ход работы

1. Пишем ТЗ “как надо”
 - a. Структура данных - 2 таблицы и поля
 - b. Все представления - 2ой объект
 - c. Все модульные сетки, учитывая второй объект
 - d. Сценарии работы
 - i. Особо проработать исключения, в частности удаление объектов, с которыми связаны другие
 - ii. Добавление объекта, ссылающегося на несуществующих
 - iii. Редактирование объекта с нарушением связности
 - iv. Удаление файла при удалении записи.

- v. Замена файла при редактировании.
 - vi. Возможность удалить файл при редактировании.
 - e. Требования к безопасности
 - i. Добавляем файлы
2. Пишем код и собираем решение на github, в протокол ссылку на проект

3. Записать скринкаст как все это работает

Работа сдается в виде протокола:

- ТЗ
- Ссылка на github (дамп базы там же)
- Скринкаст

Сдавать как ОТДЕЛЬНЫЙ ПРОЕКТ, копия предыдущего

4 и 5 ЛР оцениваются и сдаются отдельно.

Лабораторная работа 6. Собственная регистрация, поиск, авторизация через таблицу users, внешняя авторизация, внешняя интеграция.

Добавляем к 5 ЛР новые функции:

- поиск по обеим сущностям (вводим слово или число, получаем список записей, в которых оно встречается, с возможностью перейти по ссылке и посмотреть связанные данные)
 - Бонус за полнотекстовый индекс
 - Бонус за приличный интерфейс
 - Бонус за аякс
 - Бонус за вменяемо реализованную сортировку результатов по релевантности
- регистрацию ручную
 - Проверка что логин не занят
 - Валидация
 - Проверка что пароль соответствует требованиям безопасности
 - Создание пользователя в БД
 - Пароль в открытом виде не храним
 - Бонус за
 - Капчу
 - Приличный интерфейс
 - Аякс
 - Авторизацию через VK/FB, как внешний источник авторизации
- интеграцию с любым внешним сервисом по вашему выбору. Критерий -- ваш код лезет наружу и получает данные для себя, показывает их в личном кабинете пользователя
 - любым сервисом с tech.yandex.ru
 - рекомендую Спеллчекер, Предиктор, Карты, Алиса
 - любым государственным сервисом с открытыми данными
 - рекомендую ФИАС
 - <https://data.gov.ru/>
 - Dadata.ru
 - Сервисы с открытым API
 - bitrix24 — получить список задач по номеру пользователя
 - amoCRM — получить список дел на сегодня по номеру пользователя

Ход работы

1. Пишем ТЗ “как надо” - основное внимание на описание сценариев и сеток
2. Пишем код и собираем решение на github, в протокол ссылку на проект
 - a. ВАЖНО: уберите из проекта код, не относящийся к 6 ЛР
3. Записать скринкаст как все это работает

Работа сдается в виде протокола:

- ТЗ
- Ссылка на github (дамп базы там же)
 - ВАЖНО: уберите из проекта код, не относящийся к 7 ЛР
- Скринкаст

Сдавать как ОТДЕЛЬНЫЙ ПРОЕКТ, копия предыдущего
4, 5, 6 ЛР оцениваются и сдаются отдельно.

Лабораторная работа 7. Углубленная работа с данными

Добавляем к пред. Разработке еще 5 таблиц, придумываем 6 отчетов, реализуем систему отчетов.
Добавляем логирование отчетов пользователя

Добавляем к 6 ЛР 5 новых таблицы с данными:

- справочники, реестры и тп.
- например если были “страны и города”, можно добавить:
 - Визовые отношения (из какой страны в какую можно без визы и какие требования к визе)
 - Туристические маршруты
 - Ключевые достопримечательности
 - Консульства и посольства
 - Крупнейшие реки и где они протекают
- CRUD сделать на основе того что было в 5 ЛР (без копипаста)

Отчеты = хорошо представленные результаты запроса к БД. Отчет должен быть параметризован. например:

- задать пару городов и вывести подробно маршруты и достопримечательности, которые по нему можно получить, консульства рядом, реки, которые можно посмотреть
- подробная информация об отношениях между 2 странами
- подробная информация о реке (страны и города, где она протекает, с деталями по ним)
- “какие туристические маршруты и достопримечательности я могу посмотреть, отправившись на катере из города А?”
- какие города и какими способами можно посетить за 2 дня из города М?

Добавляем логирование действия пользователя: что делал, какие страницы посещал, какие изменения были внесены в БД.

15 баллов
СО

Лабораторная работа 8. Переделываем результат 5 ЛР (CRUD 2 таблиц) на ajax и работу в режиме single page application.

20 баллов
АШ/АЕ

Весенний семестр 2020/2021

Выполнено 4 лабы = пятерка

Выполнено 3 лабы = четверка

Выполнено 2 лабы = тройка

Выполнена 1 лаба = 0 баллов

Выполнено 0 лаб = 0 баллов

Семестр 2. Лабораторная работа 1

Познакомиться с 1С-Битрикс (БУС = Битрикс: Управление сайтом).

Что делать:

1 - зарегистрироваться на сайте 1c-bitrix.ru

2 - пройти курс "[Установка и настройка](#)"

3 - установить и настроить БУС к себе на локальный сервер/на хостинг, если есть, можно на XAMPP, можно на Веб-окружение.

Ставьте любое тиражное решение. Должно открыться и заработать.

Оценка времени: 4 часа с перерывом на кофе и прогулку в парке.

Семестр 2. Лабораторная работа 2

На локальной копии Битрикса начать проходить 1-ый курс Академии Битрикса

(http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&INDEX=Y)

- Пройти урок "1. Установка платформы" (~1 час)
- Пройти урок "2. Структура Bitrix Framework" (~2 часа)
- Пройти урок "3. Интеграция HTML шаблона" (~4 часа)
- Пройти урок "4.1 Компоненты. Часть 1" (~6 часов)

Принести на занятие, показать преподавателю, быть готовым ответить на вопросы “как” и “почему”. Если нет возможности принести ноутбук — загружать видео в папки старого семестра.

Семестр 2. Лабораторная работа 3

На локальной копии Битрикса начать проходить 1-ый курс Академии Битрикса

(http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&INDEX=Y)

- Пройти урок "4.2 Компоненты. Часть 2" (~4 часа)
- Пройти урок "5. Информационные блоки - обзорно, настройки" (~20 минут)
- Пройти урок "6. Вывод динамической информации" (~6 часов)
- Пройти урок "7. Кеширование. Основы работы." (~30 минут)

Принести на занятие, показать преподавателю, быть готовым ответить на вопросы “как” и “почему”. Если нет возможности принести ноутбук — загружать видео в папки старого семестра.

Семестр 2. Лабораторная работа 4

На локальной копии Битрикса начать проходить 1-ый курс Академии Битрикса

(http://academy.1c-bitrix.ru/education/index.php?COURSE_ID=65&INDEX=Y)

- Пройти урок "8. Комплексные компоненты" (~3 часа)
- Пройти урок "9. ЧПУ, "эрмитаж", постраничная навигация" (~2 часа)
- Пройти урок "10. Поиск" (~2 часа)
- Пройти урок "11. Полезные возможности" (~3 часа)

Принести на занятие, показать преподавателю, быть готовым ответить на вопросы “как” и “почему”. Если нет возможности принести ноутбук — загружать видео в папки старого семестра.