Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шутов Олег Леонтьевич
Должность: Ректор
Дата подписания: 16.11.2026 14:55:24
Уникальный программный ключ:
6892313c2153d214b87fca0fd68c13fa12d41989

Разработка программно-информационных систем
09.03.04 Программная инженерия
2023 год набора

Приложение В

к основной профессиональной образовательной программе
по направлению подготовки 09.03.04 Программная инженерия,
утвержденной приказом от 07.12.2022 г. № 1-О
(в редакции приказа от 02.11.2026 г. № 87-О)

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КУБАНСКИЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ»
(АНОО ВО «КИПО»)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.В.04 Web-программирование

Направление подготовки

09.03.04 Программная инженерия

Направленность (профиль)

Разработка программно-информационных систем

Уровень высшего образования

Бакалавриат

Квалификация

Бакалавр

Форма обучения

заочная

Год набора

2023

Рабочая программа дисциплины (модуля) Б1.В.04 «Web-программирование» предназначена для реализации основной профессиональной образовательной программы высшего образования.

Составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.04 Программная инженерия (Приказ Минобрнауки России от 19.09.2017 г. № 920 зарегистрирован в Минюсте России от 16.10.2017 г. № 48546).

СОДЕРЖАНИЕ

1 Цели и задачи изучения дисциплины(модуля)
1.1 Цель освоения дисциплины (модуля)
1.2 Задачи дисциплины (модуля)
1.3 Место дисциплины (модуля) в структуре образовательной программы
1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы
2. Структура и содержание дисциплины (модуля)
2.1 Распределение трудоёмкости дисциплины (модуля) по видам работ
2.2 Содержание дисциплины (модуля)
2.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
3. Образовательные технологии, применяемые при освоении дисциплины (модуля)
4. Фонды оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)
4.1. Структура оценочных средств для текущего контроля и промежуточной аттестации по дисциплине (модулю)
4.2. Типовые задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации по дисциплине (модулю)
5. Методические указания для обучающихся по освоению дисциплины (модуля)
6. Материально-техническое и учебно-методическое обеспечение по дисциплине (модулю)

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Целью изучения дисциплины Б1.В.04 «Web-программирование» является овладеть основными принципами разработки проектов на основе веб-технологий на языке программирования Python.

1.2 Задачи дисциплины

изучить архитектуру современных веб-приложений (модель MVC), принципы работы протокола HTTP, основные методы разработки клиентской и серверной частей; освоить современные фреймворки для веб-разработки на языке Python (Flask, Django), шаблонизаторы (Jinja2) и технологии объектно-реляционного отображения (SQLAlchemy); ознакомиться с содержанием Единого реестра российских программ для электронных вычислительных машин и баз данных в части, касающейся веб-технологий и инструментов разработки;

научиться разрабатывать технические спецификации на веб-компоненты и их взаимодействие (включая REST API), проектировать структуру веб-приложения и базы данных, реализовывать серверную логику обработки HTTP-запросов, создавать динамические веб-страницы с использованием шаблонов, внедрять системы аутентификации и разграничения прав доступа, применять ORM для работы с базами данных, а также осуществлять отладку и тестирование разработанных веб-приложений;

получить практический опыт разработки и публикации полноценного веб-приложения на языке Python с использованием веб-фреймворков (Flask/Django), включая проектирование архитектуры, интеграцию с базами данных, реализацию пользовательских интерфейсов на основе шаблонов, настройку маршрутизации, обеспечение безопасности (аутентификация, защита от типовых уязвимостей), а также опыт развертывания готового приложения на удаленной платформе (например, PythonAnywhere, Heroku) и документирования разработанных компонентов в соответствии с требованиями технической спецификации.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Б1.В.04 «Web-программирование» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» учебного плана.

В соответствии с рабочим учебным планом дисциплина изучается на 3,4 курсе по заочной форме обучения.

Вид промежуточной аттестации: зачет, экзамен.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенции	Код и наименование индикаторов достижения результатов обучения по дисциплине	Планируемые результаты обучения
ПК-8. Способен разрабатывать технические спецификации на программные компоненты и их взаимодействие	**ПК-8.1. Знает:** языки формализации функциональных спецификаций, стандарты и требования к оформлению технической документации на программные компоненты и их взаимодействие. **ПК-8.2. Умеет:** разрабатывать	**Знать:** основные языки и нотации для описания функциональных спецификаций (UML, IDEF0); структуру и содержание технической спецификации на программный компонент (API, модуль, сервис); требования к оформлению документации на web-компоненты и их интерфейсы.

Код и наименование компетенции	Код и наименование индикаторов достижения результатов обучения по дисциплине	Планируемые результаты обучения
	технические спецификации на программные компоненты и их взаимодействие, выбирать средства реализации требований к программному обеспечению. **ПК-8.3. Имеет** практический опыт**:** применения основных методов проектирования программного обеспечения для разработки и согласования технических спецификаций.	**Уметь:** анализировать требования к web-приложению и формализовывать их в виде технических спецификаций; разрабатывать спецификации на программные интерфейсы (API) для взаимодействия компонентов web-приложения; выбирать технологии и инструментальные средства для реализации специфицированных компонентов. **Владеть:** методами разработки и документирования архитектуры web-приложений; навыками составления и согласования технических спецификаций на программные компоненты с архитектором и разработчиками; навыками формализации требований к взаимодействию клиентской и серверной частей web-приложения.

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 9 зачетных единиц (324 час.), их распределение по видам работ представлено в таблице

Виды работ		Всего часов		
		ОФО	ОЗФО	ЗФО
Контактная работа, в том числе:		**-**	**-**	**36**
Аудиторные занятия (всего):		**-**	**-**	**36**
занятия лекционного типа		-	-	14
практические занятия		-	-	22
Иная контактная работа:				
Контрольная работа		-	-	-
Курсовая работа		-	-	-
Самостоятельная работа, в том числе:		**-**	**-**	**275**
Самоподготовка по темам (разделам) дисциплины		-	-	235
Подготовка к текущему контролю		-	-	40
Контроль:		**-**	**-**	**13**
Промежуточная аттестация (экзамен)		-	-	13
Общая трудоемкость	**час.**	**-**	**-**	**324**
	в том числе контактная работа	**-**	**-**	**36**
	зач. ед	**-**	**-**	**9**

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые на 3 и 4 курсе (заочная форма обучения)

№	Наименование темы/раздела	Количество часов				
		Всего	В том числе в виде практической подготовки	Аудиторная работа		Внеаудиторная работа (СР)
				Л	ПЗ	
1.	**Основы программирования**. 1.1. Простые алгоритмы 1.2. Простые конструкции 1.3. Массивы 1.4. Функции 1.5. Двумерные массивы 1.6. Массивы и строки 1.7. Рекурсия 1.8. Классы 1.9. Паттерны программирования	15	2	1	2	12
2.	**Веб-разработка с применением концепции MVC.** 2.1 Методы Веб-разработки с применением концепции MVC	22	1	1	1	20
3.	**Основы использования шаблонов.** 3.1 Создание веб-сервера и обработчиков url (flask).	33	2	1	2	30
4.	**Генерация страниц (шаблонизатор jinja2).** 4.1 Bootstrap - стилевые компоненты, css.	38	3	1	3	34
5.	**Хранение и работа с данными.** 5.1 Основные принципы хранения и работы с данными	32	2	2	2	28
5.	**Отображение данных в БД на объекты приложения (ORM).** 5.1 Flask-sqlalchemy.	26	-	1	-	25
6.	**Использование форм.** 6.1 Авторизация. 6.2 Разграничение прав доступа пользователей (flask-login).	31	1	-	1	30
7.	**Редактирование шаблонов - добавление функционала для авторизованных пользователей.** 7.1 Методы редактирования шаблонов.	23	-	1	-	22
8.	**Размещение проекта в интернете (pythonanywhere или heroku).** 8.1 Принципы и критерии размещения проекта в интернете	19	2	1	2	16
9.	**Фреймворк Django.** 9.1 Работа с шаблонами и моделями.	23	2	1	2	20
10.	**Использование библиотек Django для создания блога**. 10.1 Сложные запросы к данным.	14	1	1	1	12
11.	**Расширенные возможности шаблонного механизма**. 11.1 Принципы и задачи расширенных возможностей шаблонного механизма.	13	2	1	2	10
12.	**Выдача данных в форматах, отличных от HTML.** 12.1 Средства отладки и тестирования	11	2	1	2	8

13.	**Развертывание веб-приложений.** 13.1 Методы развертывание веб-приложений.	11	2	1	2	8
	ИТОГО по разделам дисциплины	311	22	14	22	275
	Контрольная работа	-	-	-	-	-
	Курсовая работа	-	-	-	-	-
	Промежуточная аттестация (зачет)	4	-	-	-	-
	Промежуточная аттестация (экзамен)	9	-	-	-	-
	Общая трудоемкость по дисциплине	324	22	14	22	275

Примечание: Л – лекции, ПЗ – практические занятия / семинары, СР – самостоятельная работа обучающегося

При изучении дисциплины могут применятся электронное обучение, дистанционные образовательные технологии в соответствии с ФГОС ВО.

2.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине)

Самостоятельная работа – это индивидуальная познавательная деятельность обучающегося как на аудиторных занятиях, так и во внеаудиторное время. Самостоятельная работа должна быть многогранной и иметь четко выраженную направленность на формирование конкретных компетенций.

Цель самостоятельной работы – овладение знаниями, профессиональными умениями и навыками, опытом исследовательской деятельности и обеспечение формирования профессиональных компетенций, воспитание потребности в самообразовании, ответственности и организованности, творческого подхода к решению проблем.

Самостоятельная работа обучающихся направлена на углубленное изучение разделов и тем рабочей программы. Самостоятельная работа предполагает изучение литературных источников, выполнение контрольных заданий и работ, проведение исследований разного характера. Работа основывается на анализе литературных источников и других материалов, а также реальных фактов, личных наблюдений и т.д.

Самостоятельная работа включает разнообразный комплекс видов и форм работы обучающихся:

- работа с лекционным материалом, предусматривающая проработку конспекта лекций и учебной литературы;
- поиск (подбор) и обзор литературы, электронных источников информации по заданной проблеме курса, написание реферата (доклада, эссе), исследовательской работы по заданной проблеме;
- выполнение задания по пропущенной или плохо усвоенной теме;
- изучение материала, вынесенного на самостоятельную проработку (отдельные темы, параграфы);
- подготовка к практическим занятиям;
- подготовка к промежуточной аттестации.

№ п/п	Вид учебно-методического обеспечения
1.	Методические рекомендации по самостоятельной работе обучающихся.
2.	Методические рекомендации по изучению дисциплины.
3.	Вопросы для письменного/устного собеседования, реферат, сообщение, доклад, эссе, практико-ориентированные задания, мини-кейсы, задания в виде расчетных задач, ситуационные задачи.

Задания для самостоятельной работы обучающихся по дисциплине Б1.В.04 «Web-программирование» представлены в учебно-методическом отделе.

Контроль результатов самостоятельной работы обучающихся может осуществляться в пределах времени, отведенного на обязательные учебные занятия и внеаудиторную самостоятельную работу обучающихся по дисциплине, может проходить в письменной, устной или смешанной форме.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) при изучении данной дисциплины предоставлена возможность выбора технологий обучения в зависимости от степени заболевания и осознания своей деятельности. При этом содержание программы дисциплины не изменяется, изменяются, как правило, форма обучения и образовательные технологии. Также обучающимся, имеющим инвалидность, и лицам с ограниченными возможностями здоровья созданы условия комфортного психологического климата в процессе обучения и возможности оказания помощи в установлении полноценных межличностных отношений с другими обучающимися.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В ходе изучения дисциплины предусмотрено использование следующих образовательных технологий: лекции, практические занятия, проблемное обучение, модульная технология, подготовка письменных аналитических работ, самостоятельная работа обучающихся.

Компетентностный подход в рамках преподавания дисциплины реализуется в использовании интерактивных технологий и активных методов (проектных методик, мозгового штурма, разбора конкретных ситуаций, анализа педагогических задач, педагогического эксперимента, иных форм) в сочетании с внеаудиторной работой.

Информационные технологии, применяемые при изучении дисциплины: использование информационных ресурсов, доступных в информационно-телекоммуникационной сети Интернет.

Адаптивные образовательные технологии, применяемые при изучении дисциплины – для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

4. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Б1.В.04 «Web-программирование». Материалы для проведения текущего контроля успеваемости и промежуточной аттестации размещены в фонде оценочных средств по дисциплине Б1.В.04 «Web-программирование».

4.1. Структура оценочных средств для текущего контроля и промежуточной аттестации по дисциплине (модулю)

№ п/п	Код и наименование индикаторов достижения результатов обучения по дисциплине	Результаты обучения	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	**ПК-8.1. Знает:** языки формализации функциональных спецификаций, стандарты и требования к оформлению технической документации на программные компоненты и их взаимодействие.	**Знать:** основные языки и нотации для описания функциональных спецификаций (UML, IDEF0); структуру и содержание технической спецификации на программный компонент (API, модуль, сервис); требования к оформлению документации на web-компоненты и их интерфейсы. **Уметь:** анализировать требования к web-приложению и формализовывать их в виде технических спецификаций; разрабатывать спецификации на программные интерфейсы (API) для взаимодействия компонентов web-приложения; выбирать технологии и инструментальные средства для реализации специфицированных компонентов. **Владеть:** методами разработки и документирования архитектуры web-приложений; навыками составления и согласования технических спецификаций на программные компоненты с архитектором и разработчиками; навыками формализации требований к взаимодействию клиентской и серверной частей web-приложения.	Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на зачете и экзамене
2	**ПК-8.2. Умеет:** разрабатывать технические спецификации на программные компоненты и их взаимодействие, выбирать средства реализации требований к программному обеспечению.		Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на зачете и экзамене
3	**ПК-8.3. Имеет практический опыт:** применения основных методов проектирования программного обеспечения для разработки и согласования технических спецификаций.		Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на зачете и экзамене

4.2. Типовые задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации по дисциплине (модулю)

Задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации, необходимые для оценки образовательных достижений обучающихся.

Текущий контроль успеваемости для обучающихся по заочной форме

Контрольная работа представляет собой систематическое, достаточно полное изложение авторского решения соответствующей проблемы и выполнение заданий в рамках дисциплины, которая является одним из видов текущего контроля успеваемости обучающихся заочной формы обучения.

Цели контрольной работы:

- проверка и оценка знаний обучающихся;
- закрепление практических навыков применения теоретических подходов и методов анализа на учебных примерах и задачах;
- получение информации об уровне самостоятельности и активности обучающегося, об эффективности форм и методов учебной работы.

Контрольные работы выполняются обучающимися в сроки, предусмотренные учебным планом и календарным учебным графиком.

Контрольная работа выполняется в рукописном или в печатном (компьютерном) варианте на листах формата А4 в 1 экземпляре с соблюдением установленного формата. Текст набирается шрифтом Times New Roman 12, через 1 интервал, абзацный отступ - 1,25 см, выравнивание по ширине страницы. Страница должна иметь следующие поля: левое - 25 мм, правое - 10 мм, верхнее и нижнее - 20 мм. Титульный лист содержит информацию об обучающемся выполнившем контрольную работу (ФИО обучающегося, направление подготовки, группа); наименование дисциплины; ФИО преподавателя, проверяющего работу.

Задания для контрольных работ разрабатываются преподавателем дисциплины по вариантам, которые содержат:

1) Задание в форме ответа на теоретический вопрос по теме (разделу) – объем не более 2-3 страниц;
2) Задания, составленные в форме тестов (2 задания открытого и закрытого типа, разработанные в фонде оценочных средств) или ответ на практическое задание (решить ситуацию, математическую задачу, расчетно-графическое задание и др.).

Готовая контрольная работа в электронном виде прикрепляется в электронную образовательную среду Moodle в профиль обучающегося выполнившего работу до начала сессии. Если работа в рукописном варианте, то она должна быть отсканирована и прикреплена.

Шкала и критерии оценивания контрольной работы

№ п/п	Критерии	Зачтено
Теоретический вопрос		
1	Глубина проработки материала	Основные теоретические положения по вопросу раскрыты. Имеются элементы обоснования выводов
2	Представление	Имеются элементы систематизации информации, факты применения профессиональной терминологии
3	Использование рекомендованной литературы	Основные источники рекомендованной литературы использованы
4	Грамотность изложения и качество оформления	Продемонстрирована культура речи. Соблюдены основные требования к оформлению
Выполнение тестовых заданий		

Если работа не отвечает названным критериям, выставляется оценка «не зачтено».

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет, экзамен)

Теоретические вопросы к зачету

1. Веб-разработка с применением концепции MVC.
2. Создание веб-сервера и обработчиков url (flask).
3. Генерация страниц (шаблонизатор jinja2).
4. Хранение и работа с данными. Отображение данных в БД на объекты приложения (ORM), Flask-sqlalchemy.
5. Использование форм во Flask и Django.
6. Авторизация. Разграничение прав доступа пользователей (flask-login). Редактирование шаблонов - добавление функционала для авторизованных пользователей.
7. Размещение проекта в интернете (pythonanywhere или heroku). Развертывание веб-приложений.
8. Фреймворк Django. Работа с шаблонами и моделями.
9. Использование библиотек Django для создания блога.
10. Сложные запросы к данным в Django. Расширенные возможности шаблонного механизма в Django.
11. Генерация в Django файлов в форматах, отличных от HTML.
12. Средства отладки и тестирования веб-приложений на Python.

Теоретические вопросы к экзамену

1. Понятие жизненного цикла программного обеспечения. Место технической спецификации в процессах разработки (Waterfall, Agile).
2. Классификация требований к программному обеспечению (функциональные, нефункциональные). Как они отражаются в спецификации на web-компонент?
3. Роль API в современной web-разработке. Спецификация как контракт между клиентом и сервером.
4. Языки и нотации для формализации требований: IDEF0, BPMN, UML. Применение каждого для описания web-систем.
5. Структура и содержание технического задания (ТЗ) на разработку web-приложения.
6. OpenAPI Specification (Swagger): назначение, структура, преимущества использования для документирования REST API. Генерация клиентских библиотек по спецификации.
7. Спецификация на уровне базы данных: ER-диаграмма, описание таблиц, связей, индексов и триггеров.
8. Подходы к специфицированию пользовательского интерфейса (UI): создание макетов, wireframes, UI-китов. Чем UI-спецификация отличается от технической?
9. Спецификация на интеграцию с внешними сервисами (платежные шлюзы, соцсети). Какие данные она должна содержать?
10. Инструменты для совместной работы над спецификациями (SwaggerHub, Postman, Confluence). Системы контроля версий (Git) для хранения и управления изменениями спецификаций.
11. Спецификация и тестирование: как документ может служить основой для разработки тестовых сценариев (модульное, интеграционное, API-тестирование).

12. Обработка ошибок в web-приложении: как спецификация должна описывать возможные коды ошибок и ответов HTTP.
13. Нефункциональные требования в спецификации: производительность (время ответа API), безопасность (аутентификация/авторизация, требования к паролям), масштабируемость. Язык для их формализации.
14. Версионирование API. Как спецификация помогает управлять API версиями и обеспечивать обратную совместимость.
15. Спецификация на событийно-ориентированную архитектуру: описание структуры событий, очередей и обработчиков (на примере web-сокетов или message broker).

Практические задания к экзамену (включают разработку спецификаций):

1. **Задание (API First подход):** Дано текстовое описание функционала: "Интернет-магазину нужен модуль корзины. Пользователь может добавить товар, изменить его количество, удалить товар и просмотреть содержимое корзины. Корзина привязана к пользовательской сессии."

- Разработайте спецификацию REST API для этого модуля в формате YAML (или другом формализованном виде), включающую минимум 4 endpoint'a.
- Опишите все возможные HTTP-коды ответов и их значения.
- Приложите примеры JSON-объектов для запросов и ответов.
- *(Задание проверяет ПК-8.1, ПК-8.2)*

2. **Задание (Анализ и критика):** Вам предоставлена "сырая" спецификация от заказчика (см. Приложение с заведомо плохой спецификацией — неполной, противоречивой, с ошибками). Проведите анализ спецификации, выявите все недостатки, классифицируйте их. Перепишите спецификацию в виде формального технического документа, готового для передачи в разработку. *(Задание проверяет ПК-8.2, ПК-8.3)*
3. **Задание (Спецификация и код):** Вам дан небольшой, но плохо документированный программный модуль (например, функция на Python или PHP). Ваша задача — задокументировать его, создав техническую спецификацию, в которой будет описано его назначение, алгоритм работы, все входные и выходные параметры, а также возможные побочные эффекты и исключения. *(Задание проверяет ПК-8.3)*
4. **Задание (Сквозная спецификация):** Спроектируйте и документируйте один из компонентов системы "Личный блог" (например, модуль добавления/редактирования поста). Ваша работа должна включать:

- ER-диаграмму для хранения данных поста (включая автора, дату, теги).
- Спецификацию REST API для создания, чтения, обновления и удаления (CRUD) поста.
- Описание модели данных (JSON) для передачи поста на клиентскую сторону.
- *(Задание комплексно проверяет все индикаторы ПК-8)*

Практические задания к зачету

Задание 1. Создайте программу, выводящую числа от 1 до 100.

Задание 2. Создайте программу, выводящую таблицу умножения. Выводится таблица 10x10, в каждой ячейке которой записано произведение номера столбца на номер строки.

Задание 3. Напишите скрипт, в параметрах которому будет передаваться два действительных числа $a, $b и признак операции $op, принимающий значения: add, sub, mul, div. В результате скрипт должен выводить результат операции. Например, если передано a=20&b=10&op=div, выведется: '20 / 10 = 2'.

Задание 4. Напишите скрипт выводящий ссылку с текстом '0'. При нажатии на ссылку происходит перезагрузка страницы, на вновь открывшейся странице отображается

ссылка с текстом '1'. В общем случае, если открыта страница с текстом 'n', то при переходе по ссылке открывается страница с текстом 'm', где m=n+1.

Задание 5. Создайте таблицу «Содержимое счёта» (назвать можно, например, "bill_content") со следующей структурой:

id (int, auto_increment) — суррогатный ключ

goods (varchar(255)) — товары

price (float) — цена

quantity (float) — количество

Заполните эту таблицу записями (не менее 10 записей). Напишите программу выводящую таблицу с заголовком: товар, цена, количество, стоимость. Расчёт стоимости осуществляйте в запросе.

Задание 6. Создайте таблицу «Оплаты по счетам» («payment»):

id (int, auto_increment),

bid (int) — номер счёта,

pdate (date) — дата оплаты,

summa (float) — сумма оплаты.

Заполните эту таблицу так, чтобы один счёт был полностью оплачен (несколькими частичными платежами), второй был частично оплачен (несколькими платежами), третий полностью не оплачен (ни одного платежа). Выведите таблицу задолженностей по счетам.

Задание 7. Разработайте веб-приложение, отображающее страницы, хранящиеся в таблице базы данных. Структура таблицы:

1. page_id (int, auto_increment) — идентификатор страницы
2. page_name (varchar) — название страницы
3. page_text (text) — содержимое страницы
4. page_weight (int, значение по умолчанию 1) — «вес» страницы в меню сайта

Задание 8. Для работы система должна подключать файл config.php, содержащий настройки для подключения к базе данных и параметр, управляющий детализацией протоколов работы. Система должна вести протокол работы в файл .log со следующими возможными уровнями детализации (каждый больший номер включает в себя сообщения уровней с меньшими номерами): 1. Ошибки: ошибки при соединении с базой данных, неверный ввод пароля при доступе к административному интерфейсу. 2. Предупреждения: передача некорректных параметров. 3. Сообщения: доступ к административному интерфейсу, добавление/изменение/удаление информации в базу данных. 4. Отладка: все обращения к скрипту.

Модуль «Система новостей». Информационная единица — новость — имеет структуру:

- Идентификатор новости;
- Дата-время размещения новости;
- Заголовок (текст не более 50 символов);
- Автор (выбор из списка авторов)
- Рубрика (выбор из списка рубрик)
- Анонс (краткий текст не более 250 символов);
- Текст новости (текст не более 65535 символов).

Модуль имеет представления:

1. Вывод блоков {Дата-время, Заголовок, Анонс} в ленту новостей, отсортированную в обратном хронологическом порядке и разбиваемую на страницы (по 10 новостей на страницу).
2. Вывод новостей по дате.
3. Вывод новостей по рубрике.

4. Вывод отдельной новости по идентификатору. Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные новости.

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Время последнего доступа к системе.

Вместо выбора из списка авторов для новостей реализуйте связь с таблицей пользователей. Система должна указывать автора автоматически, при добавлении новости аутентифицированным пользователем. Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей.

Задание 9. Для работы система должна подключать файл config.php, содержащий настройки для подключения к базе данных и параметр, управляющий детализацией протоколов работы. Система должна вести протокол работы в файл .log со следующими возможными уровнями детализации (каждый больший номер включает в себя сообщения уровней с меньшими номерами): 1. Ошибки: ошибки при соединении с базой данных, неверный ввод пароля при доступе к административному интерфейсу. 2. Предупреждения: передача некорректных параметров. 3. Сообщения: доступ к административному интерфейсу, добавление/изменение/удаление информации в базу данных. 4. Отладка: все обращения к скрипту.

Модуль "Учёт пользователей". Информационная единица — пользователь — имеет структуру:

- Логин (текст не более 20 символов);
- Фамилия (текст не более 50 символов);
- Имя (текст не более 50 символов);
- Отчество (текст не более 50 символов);
- Должность (выбор из списка должностей);
- Дата-время регистрации;

Модуль имеет представления:

1. Вывод списка пользователей с возможностью сортировки по колонкам, и разбиваемую на страницы (по 10 пользователей на страницу).

2. Вывод списка пользователей по заданной должности.

3. Вывод списка пользователей, зарегистрированных в заданном месяце.

4. Вывод карточки пользователя по заданному логину.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять записи о пользователях. Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные аутентификации хранится в отдельной таблице, связанной с таблицей пользователей и имеющей структуру:

- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- Время последнего доступа к системе.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс

суперпользователя позволяет назначать пользователям права входа в интерфейс администрирования и права суперпользователя.

Задание 10. Для работы система должна подключать файл config.php, содержащий настройки для подключения к базе данных и параметр, управляющий детализацией протоколов работы. Система должна вести протокол работы в файл .log со следующими возможными уровнями детализации (каждый больший номер включает в себя сообщения уровней с меньшими номерами): 1. Ошибки: ошибки при соединении с базой данных, неверный ввод пароля при доступе к административному интерфейсу. 2. Предупреждения: передача некорректных параметров. 3. Сообщения: доступ к административному интерфейсу, добавление/изменение/удаление информации в базу данных. 4. Отладка: все обращения к скрипту.

Модуль "Библиотека". Информационная единица — книга — имеет структуру:

- Идентификатор
- Название книги (текст не более 100 символов);
- Автор(ы) (текст не более 250 символов);
- Год издания;
- Аннотация (текст не более 65535 символов);
- Тематика (выбор из списка тематик);
- Выдана (NULL если книга не выдавалась, 0 если выдана);
- Имя графического файла, содержащего фото обложки.

Модуль имеет представления:

1. Вывод списка книг библиотеки с возможностью сортировки по колонкам. Список разбивается на страницы (по 10 книг на страницу).

2. Список доступных (не выданных) книг.

3. Вывод карточки книги.

4. Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять записи о книгах.

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные аутентификации хранится в отдельной таблице, имеющей структуру:

- Идентификатор;
- Логин;
- ФИО пользователя;
- Хеш пароля;
- Наличие прав суперпользователя;
- Время последнего доступа к системе.

В таблице записей о книгах модифицируйте атрибут "Выдана", чтобы он имел значение NULL, если книга никому не выдана в настоящий момент, и значение равное идентификатору пользователя, если книга выдана этому пользователю. Администраторский интерфейс позволяет указывать факт выдачи книги пользователю и возврата книги. Добавьте представление "Задолжники". Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять, изменять и удалять записи о пользователях, назначать права суперпользователя.

Задание 11. Для работы система должна подключать файл config.php, содержащий настройки для подключения к базе данных и параметр, управляющий детализацией протоколов работы. Система должна вести протокол работы в файл .log со следующими возможными уровнями детализации (каждый больший номер включает в себя сообщения уровней с меньшими номерами): 1. Ошибки: ошибки при соединении с базой данных, неверный ввод пароля при доступе к административному интерфейсу. 2. Предупреждения:

передача некорректных параметров. 3. Сообщения: доступ к административному интерфейсу, добавление/изменение/удаление информации в базу данных. 4. Отладка: все обращения к скрипту.

Модуль "Доска объявления". Информационная единица — объявление — имеет структуру:

- Идентификатор объявления;
- Дата-время размещения объявления;
- Заголовок (текст не более 100 символов);
- Автор (текст не более 50 символов);
- Email автора (текст не более 50 символов);
- Телефон (текст не более 20 символов);
- Текст объявления (текст не более 65535 символов).

Модуль имеет представления:

1. Вывод списка объявлений, отсортированный в обратном хронологическом порядке (по 20 объявлений на страницу).
2. Вывод объявления по идентификатору.
3. Вывод списка {Автор, Email, Телефон}.
4. Вывод объявлений по дате.

Модуль имеет открытый интерфейс, позволяющий добавлять, объявления. Также модуль имеет интерфейс администратора, позволяющий удалять объявления.

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- ФИО;
- Время последнего доступа к системе.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей.

Задание 12. Для работы система должна подключать файл config.php, содержащий настройки для подключения к базе данных и параметр, управляющий детализацией протоколов работы. Система должна вести протокол работы в файл .log со следующими возможными уровнями детализации (каждый больший номер включает в себя сообщения уровней с меньшими номерами): 1. Ошибки: ошибки при соединении с базой данных, неверный ввод пароля при доступе к административному интерфейсу. 2. Предупреждения: передача некорректных параметров. 3. Сообщения: доступ к административному интерфейсу, добавление/изменение/удаление информации в базу данных. 4. Отладка: все обращения к скрипту.

Модуль "Геоинформационная база". Информационная единица — географический объект — имеет структуру:

- Идентификатор объекта;
- Координата X (вещественное число);
- Координата Y (вещественное число);
- Название (текст не более 50 символов);
- Описание (текст не более 65535 символов).
- Координаты будем считать декартовыми.

Модуль имеет представления:

1. Вывод списка географических объектов (по 10 объектов на страницу).

2. Вывод карточки объекта.

3. Вывод списка объектов, удалённых друг от друга не дальше чем на указанное расстояние.

4. Вывод карточки объекта, ближайшего к указанной точке.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные объекты.

Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Время последнего доступа к системе.

Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей.

Задание 13. Для работы система должна подключать файл config.php, содержащий настройки для подключения к базе данных и параметр, управляющий детализацией протоколов работы. Система должна вести протокол работы в файл .log со следующими возможными уровнями детализации (каждый больший номер включает в себя сообщения уровней с меньшими номерами): 1. Ошибки: ошибки при соединении с базой данных, неверный ввод пароля при доступе к административному интерфейсу. 2. Предупреждения: передача некорректных параметров. 3. Сообщения: доступ к административному интерфейсу, добавление/изменение/удаление информации в базу данных. 4. Отладка: все обращения к скрипту.

Модуль "Телефонный справочник организации". Информационная единица — телефонный номер — имеет структуру:

- Идентификатор;
- Сотрудник, ответственный за телефон (выбор из списка сотрудников);
- Отдел, которому принадлежит телефон (выбор из списка сотрудников); • Телефонный номер (текст не более 20 символов);
- Описание (текст не более 65535 символов).

Модуль имеет представления:

1. Вывод списка всех телефонных номеров, сгруппированных по отделам (по 15 записей на страницу).

2. Вывод карточки телефона (полное описание).

3. Вывод списка телефонных номеров указанного отдела.

4. Вывод списка телефонных номеров указанного ответственного сотрудника.

Модуль имеет интерфейс администратора, позволяющий добавлять, изменять и удалять отдельные записи.

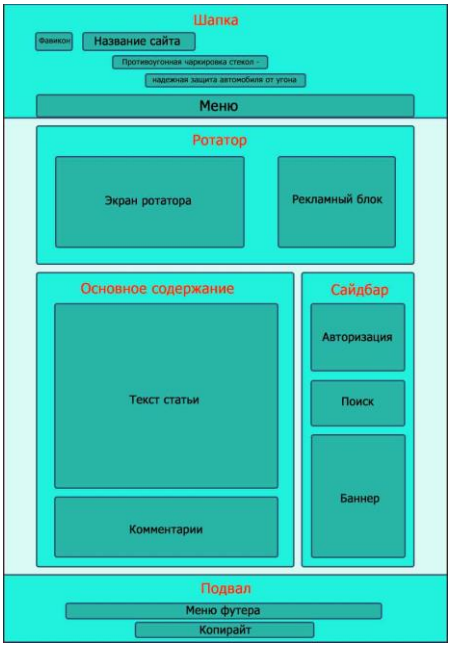
Реализовать защиту интерфейса администратора системой аутентификации пользователей. Данные пользователей хранится в таблице:

- Идентификатор;
- Логин;
- Хеш пароля;
- Наличие прав суперпользователя;
- ФИО;
- Время последнего доступа к системе.

Вместо использования перечисления сотрудников, используйте связь с таблицей пользователей. Пользователь имеет право изменять или удалять запись о телефоне, если он является ответственным за эту запись. При добавлении записи, ответственным сотрудником автоматически назначается создающий запись пользователь. Система аутентификации имеет интерфейс суперпользователя, доступный только тем пользователям, кому назначено соответствующее право. Интерфейс суперпользователя позволяет добавлять и удалять пользователей системы, а также изменять различные атрибуты существующих пользователей. Кроме того, суперпользователь имеет право создания, изменения и удаления любых записей о телефонных номерах.

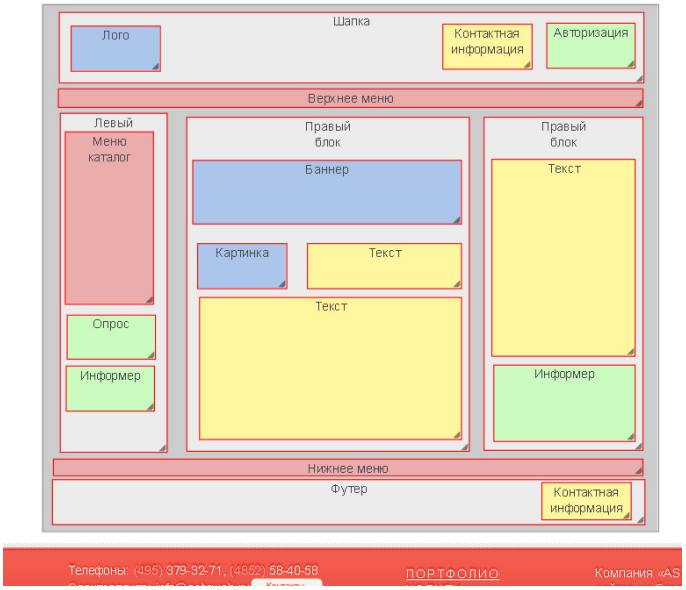
Практические задания к экзамену

Задание №1. Разработайте страницу веб-сайта «Автомобильные противоугонные системы» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в рекламном блоке макета, с применением JavaScript на странице.



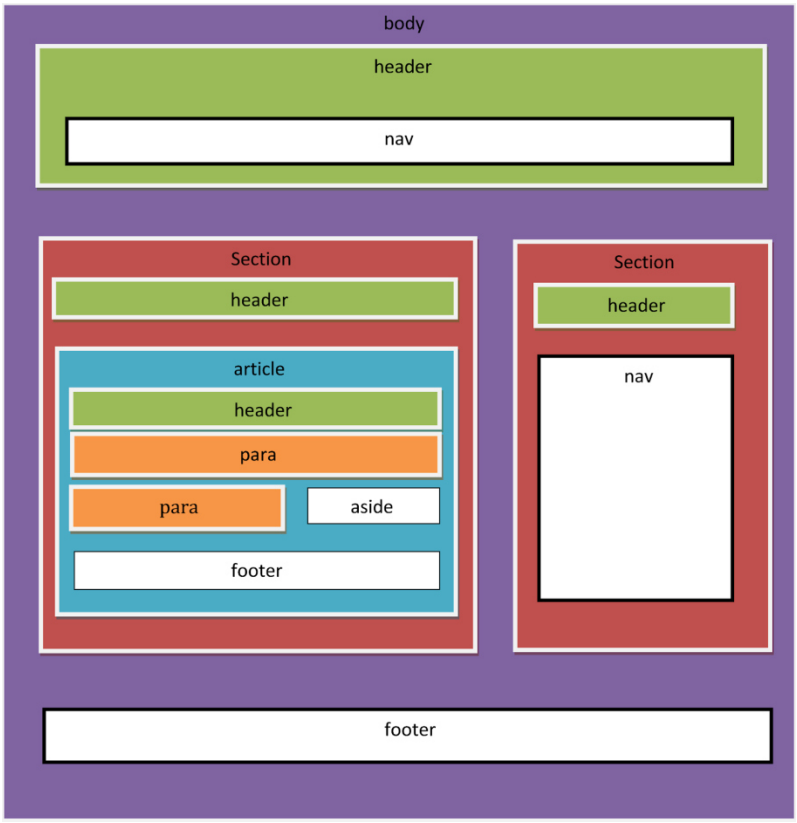
Спроектируйте базу данных предметной области «Автомобильные противоугонные системы» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №2. Разработайте страницу веб-сайта «Городская библиотека» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в шапке макета, с применением JavaScript на странице.



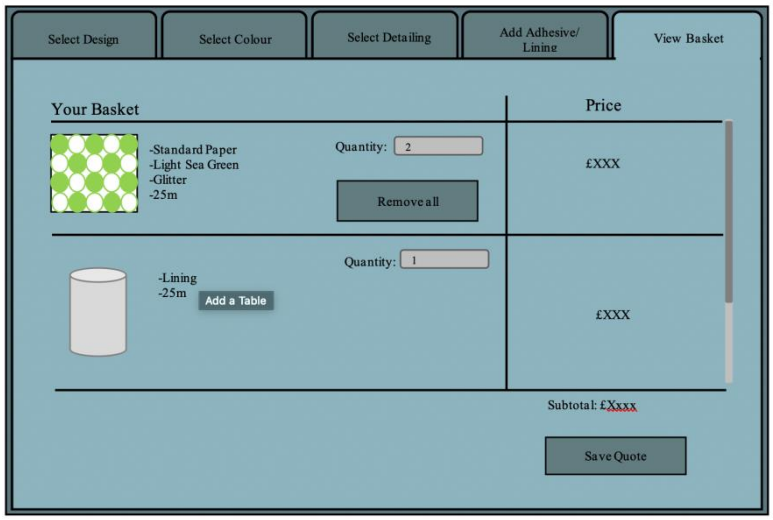
Спроектируйте базу данных предметной области «Городская библиотека» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №3. Разработайте страницу веб-сайта «Расписание» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке header макета, с применением JavaScript на странице.



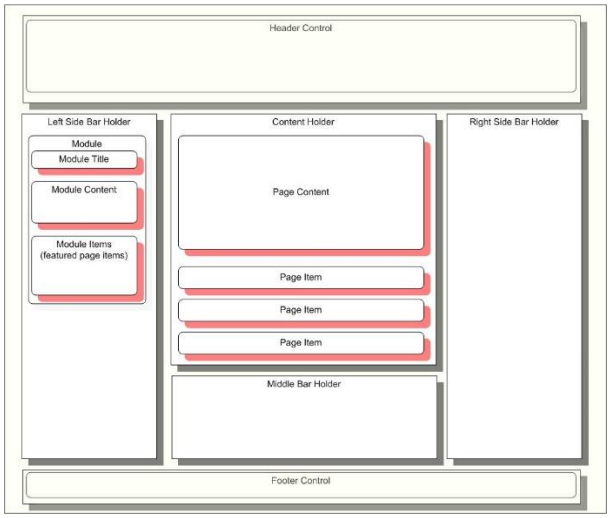
Спроектируйте базу данных предметной области «Расписание» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №4. Разработайте страницу веб-сайта «Панель администратора» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер на любой вкладке макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Панель администратора» и реализуйте ее на локальном сервере с помощью MySQL.

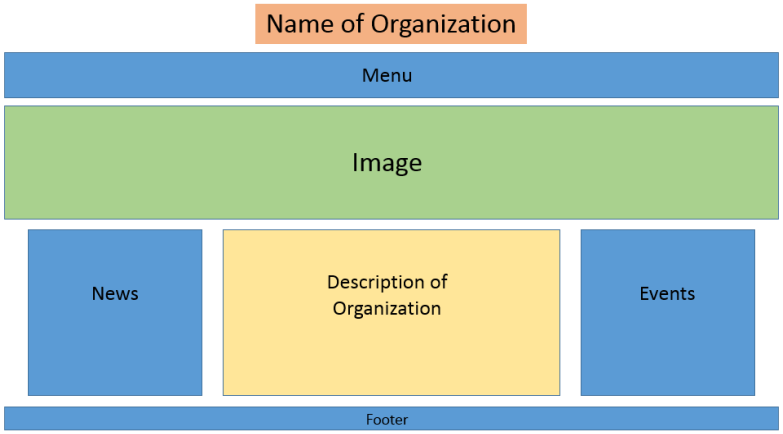
Задание №5. Разработайте страницу веб-сайта «Личный блог» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер блоке Header Control макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Личный блог» и реализуйте ее на локальном сервере с помощью MySQL.

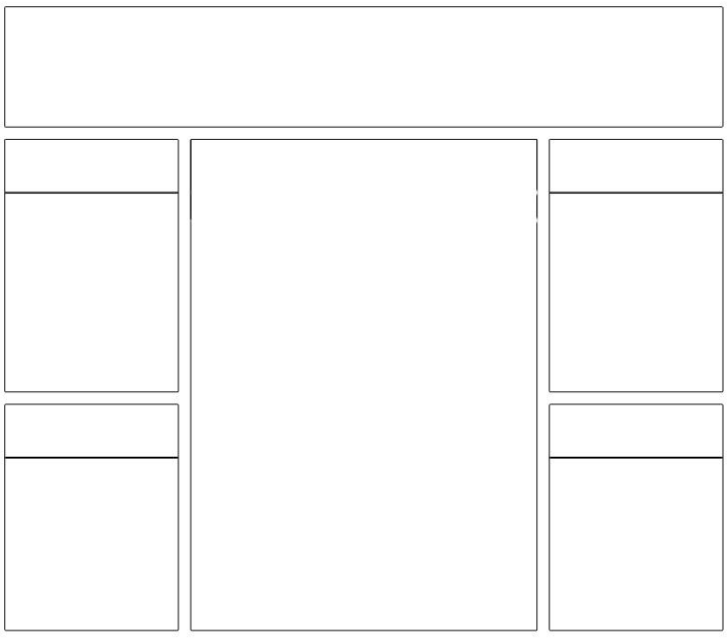
Задание №6. Разработайте страницу веб-сайта «Магазин бытовой химии» по

предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке Image макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Магазин бытовой химии» и реализуйте ее на локальном сервере с помощью MySQL.

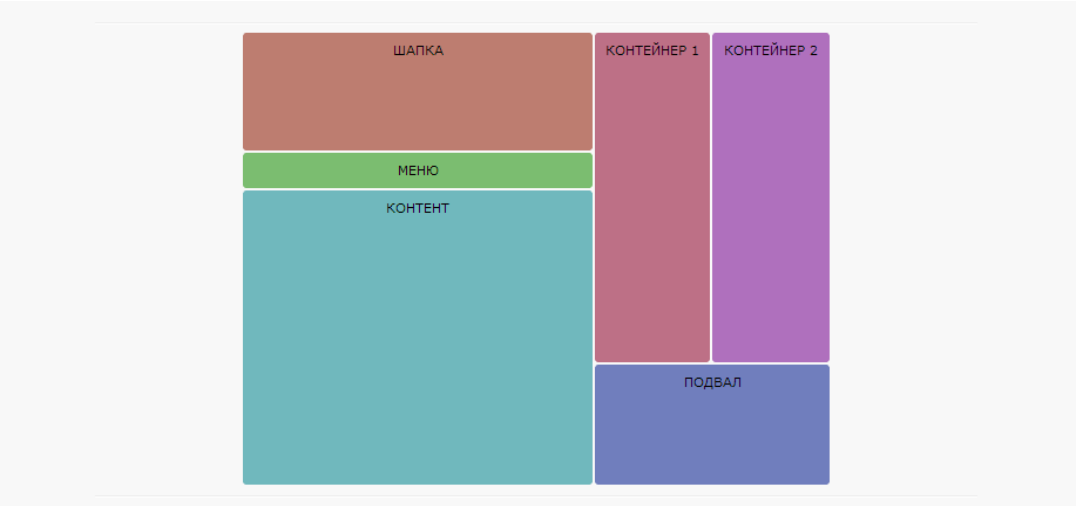
Задание №7. Разработайте страницу веб-сайта «Планировщик заданий» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в шапке макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Планировщик заданий» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №8. Разработайте страницу веб-сайта «Каталог фоновых изображений для смартфонов» по предложенному макету с применением локального сервера OpenServer на

HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированные слайдеры в контейнерах макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Каталог фоновых изображений для смартфонов» и реализуйте ее на локальном сервере с помощью MySQL.

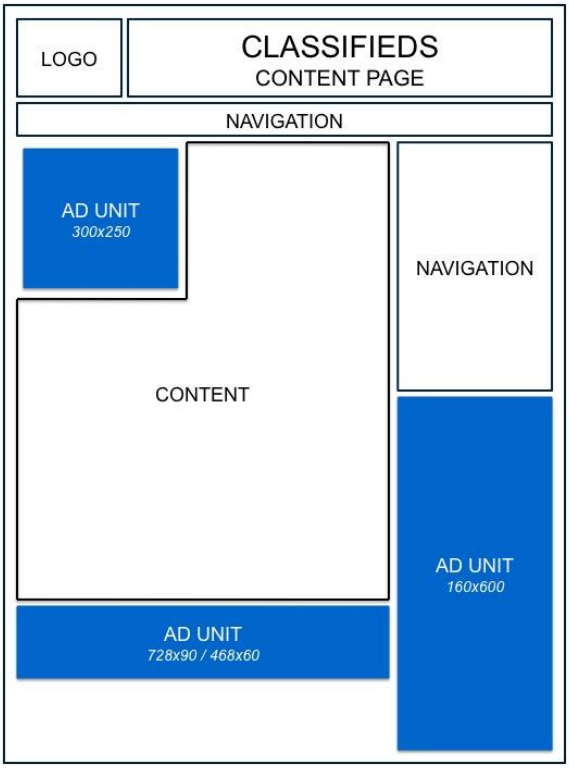
Задание №9. Разработайте страницу веб-сайта «Стоматологическая клиника» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в рекламном блоке макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Стоматологическая клиника» и реализуйте ее на локальном сервере с помощью MySQL.

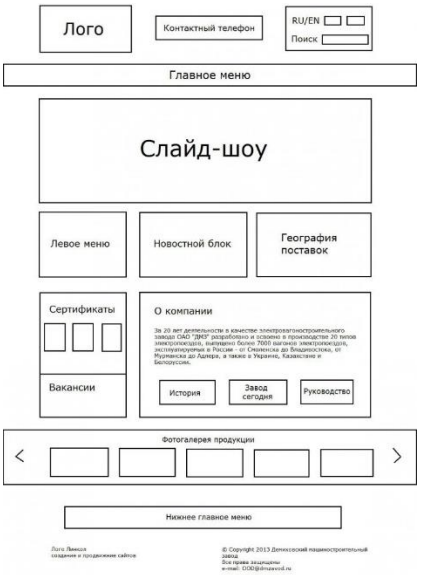
Задание №10. Разработайте страницу веб-сайта «Строительная фирма» по предложенному макету с применением локального сервера OpenServer на HTML, с

использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированные слайдеры в блоках AD UNIT макета, с применением JavaScript на странице.



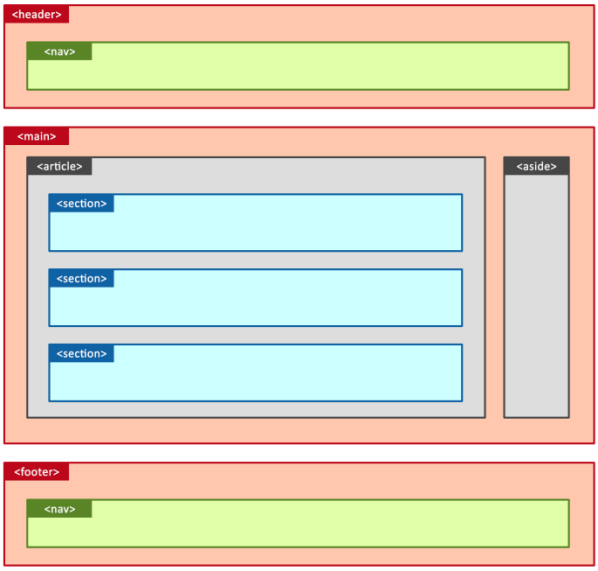
Спроектируйте базу данных предметной области «Строительная фирма» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №11. Разработайте страницу веб-сайта «Электровагоностроительный завод» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке «Слайд-шоу» макета, с применением JavaScript на странице.



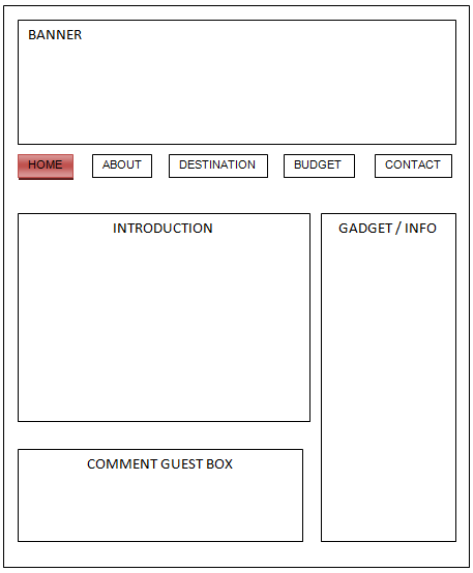
Спроектируйте базу данных предметной области «Электровагоностроительный завод» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №12. Разработайте страницу веб-сайта «Чат разработчиков» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке header макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Чат разработчиков» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №13. Разработайте страницу веб-сайта «Портал пользователей смарфонов» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке BANNER макета, с применением JavaScript на странице.



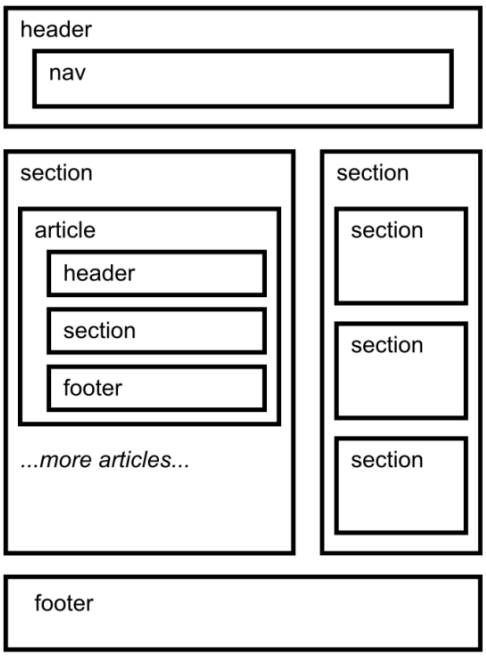
Спроектируйте базу данных предметной области «Портал пользователей смарфонов» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №14. Разработайте страницу веб-сайта «Городская библиотека» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке banner макета, с применением JavaScript на странице.



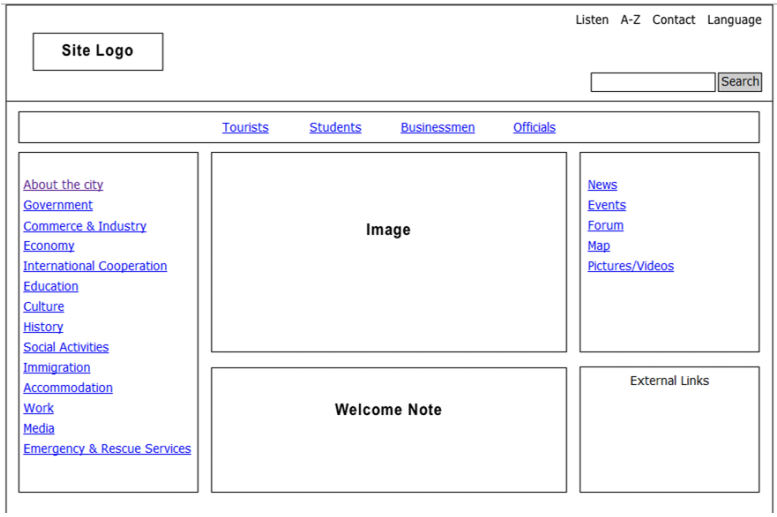
Спроектируйте базу данных предметной области «Городская библиотека» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №15. Разработайте страницу веб-сайта «Ресторан» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке header макета, с применением JavaScript на странице.



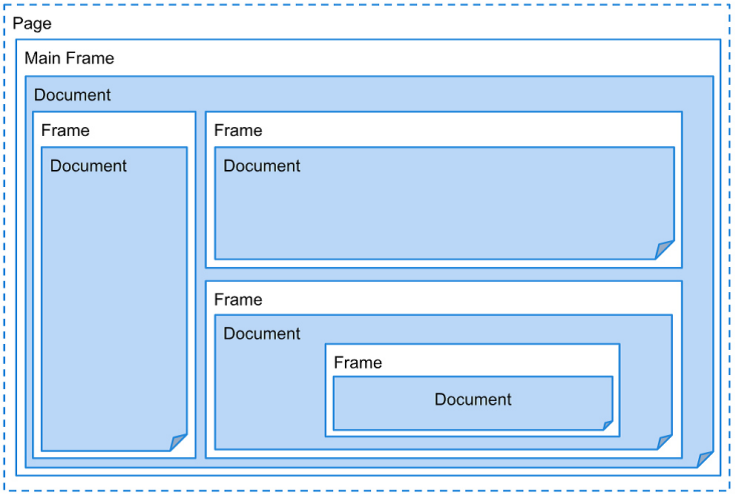
Спроектируйте базу данных предметной области «Ресторан» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №16. Разработайте страницу веб-сайта «Управляющая компания» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в шапке макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Управляющая компания» и реализуйте ее на локальном сервере с помощью MySQL.

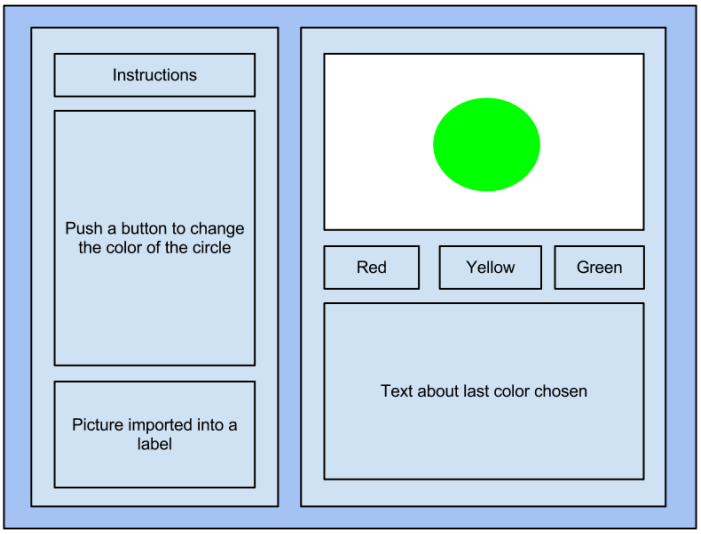
Задание №17. Разработайте страницу веб-сайта «Панель делопроизводителя» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный фон сайта, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Панель делопроизводителя» и реализуйте ее на локальном сервере с помощью MySQL.

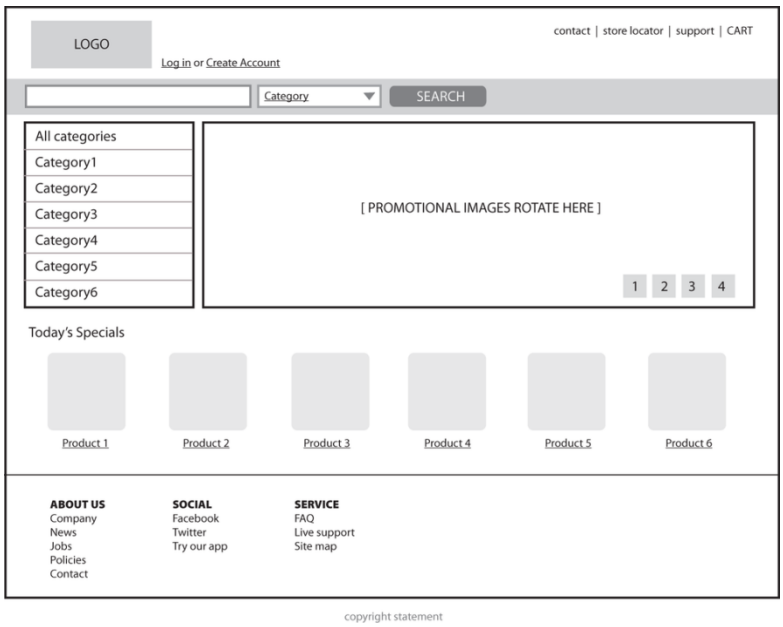
Задание №18. Разработайте страницу веб-сайта «Подбор цвета» по предложенному

макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный фон сайта, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Подбор цвета» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №19. Разработайте страницу веб-сайта «Каталог бытовой техники» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке PROMOTIONAL IMAGES ROTATE HERE макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Каталог бытовой техники» и реализуйте ее на локальном сервере с помощью MySQL.

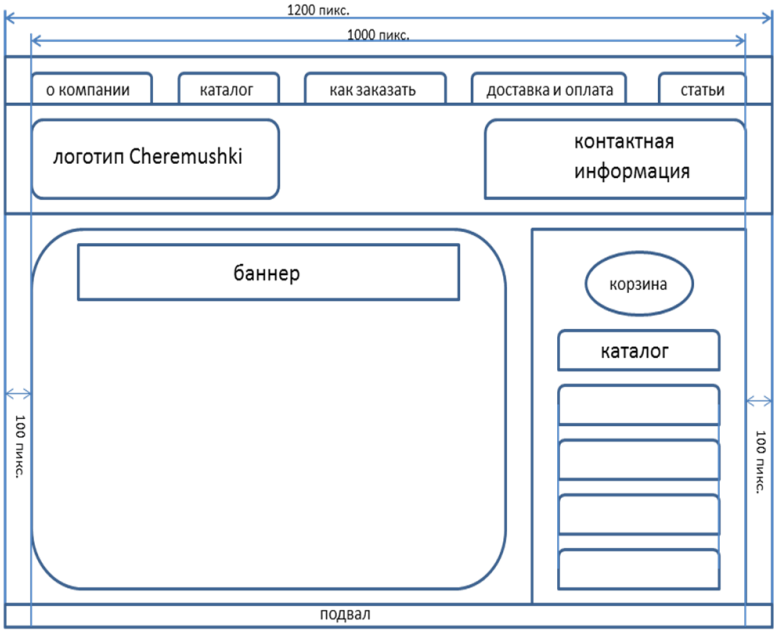
Задание №20. Разработайте страницу веб-сайта «Электронный дневник

школьника» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в рекламном блоке макета, с применением JavaScript на странице.



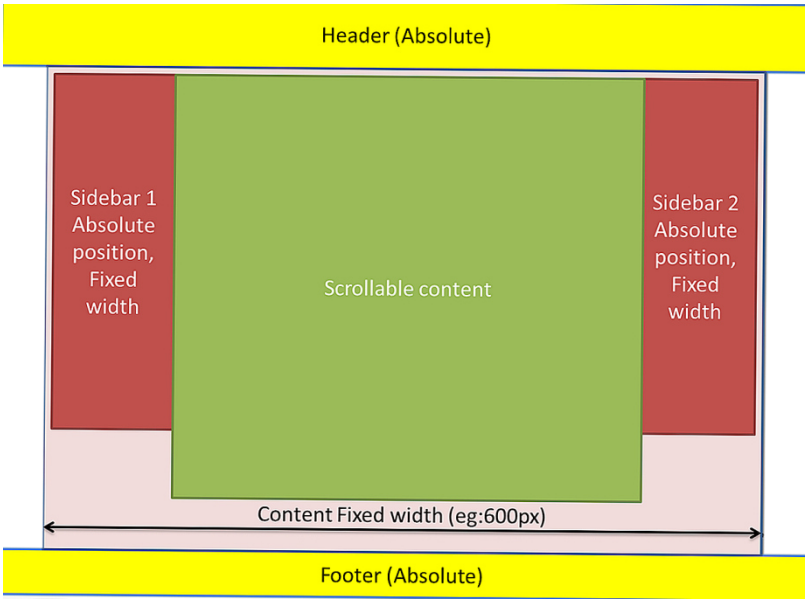
Спроектируйте базу данных предметной области «Электронный дневник школьника» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №21. Разработайте страницу веб-сайта «Продажа овощей и фруктов» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в баннере макета, с применением JavaScript на странице.



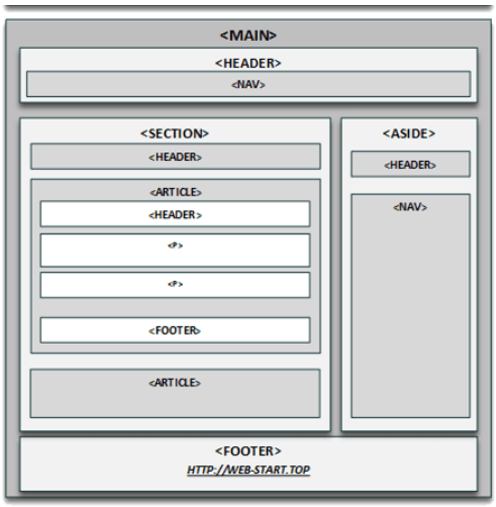
Спроектируйте базу данных предметной области «Продажа овощей и фруктов» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №22. Разработайте страницу веб-сайта «Новости» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированные слайдеры в блоках Sidebar макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Новости» и реализуйте ее на локальном сервере с помощью MySQL.

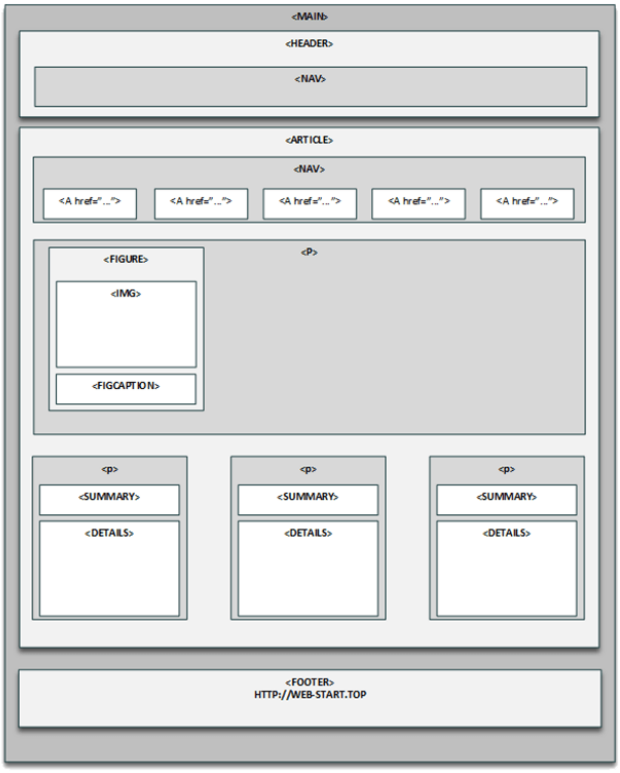
Задание №23. Разработайте страницу веб-сайта «Фитнес-клуб» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в блоке HEADER макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Фитнес-клуб» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №24. Разработайте страницу веб-сайта «Портал сайта города» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению.

Создайте анимированный слайдер в блоке HEADER макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Портал сайта города» и реализуйте ее на локальном сервере с помощью MySQL.

Задание №25. Разработайте страницу веб-сайта «Новостной портал микрорайона» по предложенному макету с применением локального сервера OpenServer на HTML, с использованием CSS, PHP. Дизайн элементов страницы выберите по своему усмотрению. Создайте анимированный слайдер в тизерных блоках макета, с применением JavaScript на странице.



Спроектируйте базу данных предметной области «Новостной портал микрорайона» и реализуйте ее на локальном сервере с помощью MySQL.

Критерии оценивания промежуточной аттестации: зачет

Оценка	Критерии оценивания по зачету
«зачтено»	заслуживает обучающийся, полностью или практически полностью освоивший знания, умения, компетенции и теоретический материал, учебные задания не оценены максимальным числом баллов, в основном сформировал практические навыки.
«не зачтено»:	заслуживает обучающийся, не освоивший знания, умения, компетенции и теоретический материал, учебные задания не выполнил, практические навыки не сформированы.

Критерии оценивания промежуточной аттестации: экзамен

Оценка	Критерии оценивания по экзамену
Высокий уровень «5» (отлично)	оценку «отлично» заслуживает обучающийся, освоивший знания, умения, компетенции и теоретический материал без пробелов; выполнивший все задания, предусмотренные учебным планом на высоком качественном уровне; практические навыки профессионального применения освоенных знаний сформированы.
Средний уровень «4» (хорошо)	оценку «хорошо» заслуживает обучающийся, практически полностью освоивший знания, умения, компетенции и теоретический материал, учебные задания не оценены максимальным числом баллов, в основном сформировал практические навыки.
Пороговый уровень «3» (удовлетворительно)	оценку «удовлетворительно» заслуживает обучающийся, частично с пробелами освоивший знания, умения, компетенции и теоретический материал, многие учебные задания либо не выполнил, либо они оценены числом баллов близким к минимальному, некоторые практические навыки не сформированы.
Минимальный уровень «2» (неудовлетворительно)	оценку «неудовлетворительно» заслуживает обучающийся, не освоивший знания, умения, компетенции и теоретический материал, учебные задания не выполнил, практические навыки не сформированы.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Методические указания для обучающихся по освоению дисциплины (модуля)

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

При проведении учебных занятий по дисциплине обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплины в форме курса, составленного на основе результатов научных исследований, проводимых институтом, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

Результат обучения считается сформированным, если теоретическое содержание курса освоено полностью; при устных собеседованиях обучающийся исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, обучающийся способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий.

Результат обучения считается несформированным, если обучающийся при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполненных заданий не соответствует установленным требованиям.

Методические указания для обучающихся по освоению дисциплины на занятиях лекционного типа

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины. Обозначаются ключевые аспекты тем, а также делаются акценты на наиболее сложные и важные положения изучаемого материала. Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям и выполнения заданий самостоятельной работы, а также к мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

В ходе лекционных занятий необходимо вести конспектирование учебного материала. Возможно ведение конспекта лекций в виде интеллект-карт.

Методические указания для обучающихся по освоению дисциплины на

занятиях практического типа

Практические (семинарские) занятия представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают все основные разделы. Основной формой проведения семинаров и практических занятий является обсуждение наиболее проблемных и сложных вопросов по отдельным темам, а также решение задач и разбор примеров и ситуаций в аудиторных условиях.

Практические (семинарские) занятия обучающихся обеспечивают:

- проверку и уточнение знаний, полученных на лекциях;
- получение умений и навыков составления докладов и сообщений, обсуждения вопросов по учебному материалу дисциплины;
- подведение итогов занятий по рейтинговой системе, согласно технологической карте дисциплины.

Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обеспечивает подготовку обучающегося к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающегося на занятиях и в качестве выполненных практических заданий и других форм текущего контроля.

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

6. Материально-техническое и учебно-методическое обеспечение по дисциплине (модулю)

Основная литература:

1. Иванова, Г. С., Технология программирования : учебник / Г. С. Иванова. — Москва : КноРус, 2026. — 333 с. — ISBN 978-5-406-15527-1. — URL: https://book.ru/book/959826— Текст : электронный.

2. Web-разработки в asp. Net web forms : учебник для вузов / С. Т. Гуляева, В. В. Миронов, Н. О. Котелина, И. И. Лавреш. — Москва : Издательство Юрайт, 2025. — 134 с. — (Высшее образование). — ISBN 978-5-534-19885-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/569218

3. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебник для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2025. — 219 с. — (Высшее образование). — ISBN 978-5-534-16300-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561176

Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС) и базы данных

Доступ к ЭБС предоставляется из любой точки, в которой имеется доступ к сети «Интернет», как на территории Института, так и вне ее (удаленный доступ).

1. Образовательная платформа «ЮРАЙТ» - URL: https://urait.ru/
2. Электронно-библиотечная система «BOOK.ru» - URL: https://www.book.ru
3. Научная электронная библиотека eLIBRARY.RU - URL: https://elibrary.ru (крупнейшая российская база научных публикаций, доступ к рефератам и полным текстам статей).
4. КиберЛенинка - URL: https://cyberleninka.ru (научная электронная библиотека открытого доступа).

Информационные справочные системы

1. Справочная правовая система «Консультант Плюс» - доступ по локальной сети с компьютеров библиотеки и компьютерных классов (актуальная база законодательства РФ, в т.ч. в сфере образования и социальной защиты).
2. Справочная правовая система «Гарант» - доступ по локальной сети с компьютеров библиотеки и компьютерных классов (актуальная база законодательства РФ, включая сферы образования и социальной защиты).

Профессиональные сообщества и IT-порталы

1. Habr - URL: https://habr.com/ (крупнейшее русскоязычное сообщество IT-специалистов, статьи, новости, обсуждения).
2. Stack Overflow - URL: https://stackoverflow.com/ (международный ресурс для программистов, вопросы и ответы).
3. GitHub - URL: https://github.com/ (платформа для хостинга кода, совместной разработки и открытых проектов).
4. CodeProject - URL: https://www.codeproject.com/ (статьи, примеры кода, обсуждения для разработчиков).
5. Microsoft Learn - URL: https://learn.microsoft.com/ (бесплатные учебные материалы и документация по продуктам Microsoft, .NET, Azure и др.).
6. MDN Web Docs - URL: https://developer.mozilla.org/ (ресурс для **веб-разработчиков с документацией по HTML, CSS, JavaScript и API).**

Научные базы данных зарубежных издательств (открытый доступ)

7. IEEE Xplore - URL: https://ieeexplore.ieee.org/ (доступ к книгам, статьям и материалам конференций по компьютерным наукам, электротехнике и информационным технологиям).
8. ACM Digital Library - URL: https://dl.acm.org/ (библиотека статей и материалов конференций Ассоциации вычислительной техники).
9. SpringerLink - URL: https://link.springer.com/ (книги и журналы издательства Springer, включая серии по информатике).
10. Wiley Online Library - URL: https://onlinelibrary.wiley.com/ (коллекция журналов и книг по направлению «Computer Science & Information Technology»).

Образовательные платформы и онлайн-курсы

11. Национальная платформа «Открытое образование» - URL: https://openedu.ru/ (курсы ведущих российских вузов).
12. Stepik - URL: https://stepik.org/ (российская образовательная платформа с курсами по программированию и информатике).
13. Intuit - URL: https://intuit.ru/ (национальный открытый университет, курсы по информационным технологиям).

Официальные органы государственной власти и управления

1. Министерство науки и высшего образования Российской Федерации - URL: https://www.minobrnauki.gov.ru/.
2. Министерство просвещения Российской Федерации - URL: https://edu.gov.ru/.

3. Министерство труда и социальной защиты Российской Федерации - URL: https://mintrud.gov.ru/.
4. Федеральная служба по надзору в сфере образования и науки (Рособрнадзор) - URL: http://obrnadzor.gov.ru/.
5. Государственная система правовой информации «Законодательство России» - URL: http://pravo.gov.ru/.

Образовательные и справочные порталы

Федеральный портал «Российское образование» - URL: http://www.edu.ru/.

Российское общество «Знание» - URL: https://znanierussia.ru/.

Справочно-информационный портал «Грамота.ру» - URL: http://gramota.ru/ (русский язык и культура речи).

Образовательный портал «Учеба» - URL: http://www.ucheba.com/.

Словари и энциклопедии на Академике - URL: https://dic.academic.ru/.

Проект Государственного института русского языка им. А.С. Пушкина «Образование на русском» - URL: https://pushkininstitute.ru/.

Комплект лицензионного и свободно распространяемого программного обеспечения:

1. LibreOffice - офисный пакет
2. PDFedit – программа для работы с pdf
3. Yandex Browser – браузер
4. Менеджер архивов
5. Libre Base – программа для работы с БД
6. Inkscape – ПО для компьютерной графики
7. DIA – ПО для блока схем и диаграмм
8. 1С: Предприятие – Учебная версия
9. GiMP - Программа обработки изображений

Перечень материально-технического обеспечения включает:

учебные аудитории для проведения занятий лекционного типа, занятий практического (семинарского) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы оснащенные компьютерной техникой с подключением к сети "Интернет" и доступом в электронную информационно-образовательную среду Института.

Наименование помещения. Перечень основного оборудования	Адрес
Учебная аудитория № 217 Учебная аудитория для проведения занятий лекционного и практического типа, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации). Оборудование: рабочее место преподавателя (1); рабочие места обучающихся (36); ноутбук с лицензионным ПО (LibreOffice) и возможностью выхода в интернет (1); мультимедийное оборудование (1); доска учебная (1); книжный шкаф (1); сплит-система(1); учебно-наглядные пособия;	350002, Краснодарский край, г. Краснодар, Центральный внутригородской округ, ул. им. Леваневского, д. 187/1

доступ в электронную информационно-образовательную среду Института.	
Аудитория № 218 Помещение для самостоятельной работы обучающихся Оборудование: рабочие места обучающихся (17); персональный компьютер с лицензионным ПО и возможностью выхода в Интернет (17); книжный шкаф (1); сплит-система (1); учебно-наглядные пособия; доступ в электронную информационно-образовательную среду Института.	350002, Краснодарский край, г. Краснодар, Центральный внутригородской округ, ул. им. Леваневского, д. 187/1