Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 31.08.2023 14:58:14
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5172742753c18b5d8

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета

«Информационные технологии»

_____/Д.Г.Демидов/

«16» мая 2022

Рабочая программа дисциплины

«Серверная веб-разработка»

Направление подготовки:

09.03.01 Информатика и вычислительная техника

Образовательная программа (профиль):

«Веб-технологии»

Год начала обучения:

2022

Уровень образования:

Бакалавриат

Квалификация (степень) выпускника:

Бакалавр

Форма обучения:

очная

Москва, 2022

Рабочая программа дисциплины «Серверная веб-разработка» составлена в соответствии с федеральным образовательным стандартом высшего образования – бакалавриата по направлению подготовки 09.03.01 Информатика и вычислительная техника.

Заведующий кафедрой «Инфокогнитивные технологии»:



к.т.н., доцент /Е.А.Пухова /

Согласовано:

Руководитель образовательной программы:



_____ /М.В.Даньшина/

Программу составили:



Преподаватель _____ / О.В.Володина/

_____ / _____ /

_____ / _____ /

_____ / _____ /

_____ / _____ /

1. Цели освоения дисциплины

Цель освоения дисциплины «Серверная веб-разработка» дать хороший общий обзор того, как работает фреймворк Laravel. Данный курс предназначен для начинающих веб-разработчиков, а также тех, кто хочет понять, для чего стоит изучить PHP-фреймворк Laravel и какую экосистему он предлагает. Веб фреймворк Ларавел (Laravel) уже многие годы считается наилучшим PHP фреймворком для разработки сайтов. Он предоставляет готовые шаблонные решения для реализации веб проектов различной сложности.

Задачи дисциплины «Серверная веб-разработка»:

Научиться настраивать окружение для локальной веб-разработки с использованием фреймворка Laravel и программной платформы Docker;

ознакомиться с HTML, Core PHP, и Advanced PHP;

изучить тонкости создания проектов и основы Laravel;

Разработать сайт на Laravel с нуля.

В результате освоения дисциплины у обучающихся формируются следующие знания и навыки:

Знают возможности существующей программно-технической архитектуры фреймворка Laravel;

обладают навыками работы с современным и перспективным фреймворком для разработки серверных веб приложений;

применяют методологии и технологии проектирования и использования баз данных;

знают принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина относится к числу учебных дисциплин обязательной части.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Мультимедиа-технологии;
- Администрирование серверов;
- Основы программирования;
- Основы веб-технологий;
- Веб-разработка;
- Базы данных.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций.

Код компетенции	Наименование компетенции	Индикаторы планируемых результатов обучения по дисциплине
ПК-1	Способен разрабатывать требования и проектировать программное обеспечение	ПК-1.1. Знать: методологии и технологии проектирования и использования баз данных; методы и средства проектирования баз данных. ПК-1.2. Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-1.3. Владеть: современным инструментарием и средами разработки программного кода.
ПК-2	Способен осуществлять	ПК-2.2. Уметь:

	концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	исполнять ручные тесты. ПК-2.3. Владеть: средствами автоматизации проектирования ПО.
ПК-4	Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-4.2. Уметь: разрабатывать справочники по интерфейсам прикладного программирования; разрабатывать руководство по языку программирования.
ПК-5	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов	ПК-5.1. Знать: языки веб-разработки. ПК-5.3. Владеть: языками веб-разработки.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетных единицы, т.е. 180 академических часа (из них 108 часов – самостоятельная работа студентов).

Разделы дисциплины изучаются на втором курсе в третьем семестре, форма промежуточной аттестации – экзамен, курсовой проект.

Содержание дисциплины

Раздел 1. Ознакомление. Настройка окружения

Тема 1. Настройка окружения

Рассмотрение особенностей фреймворка. Установка Docker, PHP, Composer, Laravel.

Раздел 2. Основы работы с фреймворком

Тема 1. Маршрутизация

Простейшие маршруты Laravel принимают URI и замыкание, обеспечивая нетрудоемкий и выразительный метод определения маршрутов и поведения без сложных конфигурационных файлов маршрутизации. Создание сайта с использованием Blade и Router (информационный сайт компании. О нас, Контакты)

Тема 2. Шаблонизатор Blade

Blade – это простой, но мощный движок шаблонов, входящий в состав Laravel. Создание сайта с использованием Blade и Router (информационный сайт компании. О нас, Контакты)

Тема 3. Контроллеры, экшены

Вместо того чтобы определять всю логику обработки запросов как замыкания в файлах маршрутов, вы можете организовать это поведение с помощью классов «контроллеров». Контроллеры могут сгруппировать связанную логику обработки запросов в один класс. Добавление на сайт динамических страниц (например с данными из json файла "Новости компании" и "Галерея") и формы (с валидацией на backend) с записью в json.

Тема 4. Обработка запроса (Request, валидация). Формирование ответа (Response, json).

Класс Illuminate\Http\Request Laravel предлагает объектно-ориентированный способ взаимодействия с текущим HTTP-запросом,

обрабатываемым вашим приложением, а также извлечение входных данных, файлов Cookies и файлов, отправленных вместе с запросом. Все маршруты и контроллеры должны возвращать ответ, который будет отправлен обратно в браузер пользователя. Laravel предлагает несколько разных способов вернуть ответы. Самый простой ответ – это возврат строки из маршрута или контроллера. Фреймворк автоматически преобразует строку в полный HTTP-ответ. Добавление на сайт динамических страниц (например с данными из json файла "Новости компании" и "Галерея") и формы (с валидацией на backend) с записью в json.

Тема 5. Работа с формами (CSRF).

Laravel автоматически генерирует «токен» CSRF для каждой активной пользовательской сессии, управляемой приложением. Этот токен используется для проверки того, что аутентифицированный пользователь действительно является лицом, выполняющим запросы к приложению. Поскольку этот токен хранится в сессии пользователя и изменяется каждый раз при повторном создании сессии, вредоносное приложение не может получить к нему доступ. Добавление на сайт динамических страниц (например с данными из json файла "Новости компании" и "Галерея") и формы (с валидацией на backend) с записью в json.

Раздел 3. Работа над сайтом

Тема 1. Работа с БД. ORM-система Eloquent.

Переработка динамических страниц ("Новости", "Галерея") на работу с БД и пагинацией, формы CRUD. Laravel содержит ORM-библиотеку Eloquent, предоставляющую способ работы с базой данных, который часто удобнее обычного построителя запросов. При использовании Eloquent каждая таблица БД имеет соответствующую «Модель», которая используется для взаимодействия с этой таблицей. Помимо получения записей из таблицы БД, модели Eloquent также позволяют вставлять, обновлять и удалять записи из таблицы. Переписать существующую работу с БД на Модели. Добавить новую

логику в сайт с зависимостями между моделями (Например Категория -> Новость -> Комментарии).

Тема 2. Аутентификация и авторизация.

По своей сути средства аутентификации Laravel состоят из «охранников» и «провайдеров». Охранники определяют, как пользователи проходят проверку подлинности для каждого запроса. Например, Laravel поставляется с охранником session, который поддерживает состояние, используя хранилище сессий и файлы Cookies.

Провайдеры определяют, как пользователи извлекаются из вашего постоянного хранилища. Laravel поставляется с поддержкой получения пользователей с помощью Eloquent и построителя запросов к базе данных. Однако вы можете определить дополнительных провайдеров, если это необходимо для вашего приложения.

Добавить на сайт систему пользователей, ограничения доступа, роли, формы авторизации и регистрации, интеграция в CRUD контроллеры (Добавить 2 роли, пользователь и модератор. Пользователь может только читать новости и комментировать их, Модератору доступны все действия с новостями)

Тема 3. Рассылка email. Broadcasting, events, notifications.

Laravel обеспечивает поддержку отправки уведомлений по различным каналам доставки, включая электронную почту, SMS (через Vonage, бывший Nexmo) и Slack. Кроме того, сообществом было создано множество каналов уведомлений для отправки уведомлений по десяткам различных каналов! Уведомления также могут храниться в базе данных, поэтому они могут быть отображены в вашем веб-интерфейсе.

Добавить на сайт модерацию комментариев с уведомлением модератора по Email о новом комментарии и кнопками (принять\отклонить). Внедрение на сайт Pusher с выводом уведомления о новых новостях и комментариях.

Тема 4. Кэширование данных, очереди.

Некоторые задачи по извлечению или обработке данных, выполняемые вашим приложением, могут потребовать больших ресурсов ЦП или занять несколько секунд. В этом случае извлеченные данные обычно кешируют на некоторое время, чтобы их можно было быстро извлечь при последующих запросах тех же данных. Кешированные данные обычно хранятся в хранилище с быстрым доступом данных, например, Memcached или Redis. Добавить на сайт кеширование Новостей и Комментариев.

При создании веб-приложения у вас могут быть некоторые задачи, такие как синтаксический анализ и сохранение загруженного файла CSV, выполнение которых во время обычного веб-запроса занимает слишком много времени. К счастью, Laravel позволяет легко создавать задания (jobs) в очереди (queue), которые могут обрабатываться в фоновом режиме. Перемещая трудоемкие задания в очередь и выполняя их в фоне, ваше приложение может быстрее обрабатывать веб-запросы и быстрее отвечать клиенту. Подключить рассылку увеломлений Pusher к очереди.

Тема 5. Планировщик задач.

Планировщик команд Laravel предлагает новый подход к управлению запланированными задачами на вашем сервере. Планировщик позволяет вам быстро и выразительно определять расписание команд в самом приложении Laravel. Добавить на сайт запланированное задание с рассылкой на email модераторов статистики использования сайта за день (кол-во просмотров новостей, новые комментарии).

Тема 6. Backend API.

Добавить на сайт backend api для работы с новостями (CRUD).

Тематика семинарских/практических и лабораторных занятий

Раздел 2. Основы работы с фреймворком

Лабораторная работа №1: «Создание макета страниц сайта»

Лабораторная работа №2: «Вывод данных, полученных из json-файла»

Лабораторная работа №3: «Регистрация пользователей»

Лабораторная работа №4: «Реализуем функционал для работы с новостями на сайте»

Лабораторная работа №5: «Пагинация на странице.»

Раздел 3. Работа над сайтом

Лабораторная работа №1: «CRUD-действия с новостями»

Лабораторная работа №2: «Авторизация/аутентификация пользователя в приложении»

Лабораторная работа №3: «Добавляем роли пользователям. Работаем с политиками»

Лабораторная работа №4: «Настраиваем e-mail рассылку»

Лабораторная работа №5: «Добавление рассылки в очередь задач»

Лабораторная работа №6: «Backend API»

5. Образовательные технологии

Методика преподавания дисциплины «Серверная веб-разработка» предусматривает использование следующих форм проведения занятий:

- выполнение лабораторных работ;
- индивидуальные и групповые консультации студентов преподавателем;
- посещение профильных конференций и работа на мастер-классах экспертом и специалистов веб-технологиях, веб-разработке, Интернет-маркетинге и других профессиональных областях.

Самостоятельная внеаудиторная работа студентов состоит из подготовки к выполнению и защите лабораторных работ, а также подготовки к промежуточной аттестации во время экзаменационной сессии.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов.

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- Лабораторные работы,
- Экзамен,
- Курсовой проект.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Серверная веб-разработка»

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

В результате освоения дисциплины формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ПК-1	Способен разрабатывать требования и проектировать программное обеспечение
ПК-2	Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности
ПК-4	Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы
ПК-5	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплины в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины, описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение				
ПК-1.1. Знать: методологии и технологии проектирования и использования баз данных; методы и средства проектирования баз данных. ПК-1.2. Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

программного обеспечения, структур данных, баз данных, программных интерфейсов. ПК-1.3. Владеть: современным инструментарием и средами разработки программного кода.		при оперировании знаниями при их переносе на новые ситуации.		
ПК-2**.** Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности				
ПК-2.2. Уметь: исполнять ручные тесты. ПК-2.3. Владеть: средствами автоматизации проектирования ПО.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

		новые ситуации.		
ПК-4. Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы				
ПК-4.2. Уметь: разрабатывать справочники по интерфейсам прикладного программирования; разрабатывать руководство по языку программирования.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.
ПК-5. Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов				
ПК-5.1. Знать: языки веб-разработки. ПК-5.3. Владеть:	Обучающийся демонстрирует полное отсутствие или	Обучающийся демонстрирует неполное соответствие следующих знаний,	Обучающийся демонстрирует частичное соответствие следующих знаний,	Обучающийся демонстрирует полное соответствие следующих знаний,

языками веб-разработки.	недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине, при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
Отлично	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при переносе знаний и умений на новые, нестандартные задачи.
Хорошо	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. В то же время при ответе допускает несущественные погрешности, задачи решает с недочетами, не влияющими на общий ход решения.
Удовлетворительно	Выполнены все **обязательные условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины. Студент демонстрирует соответствие знаний, умений, навыков показателям, приведенным в таблицах, оперирует приобретенными знаниями, умениями, навыками. Но показывает неглубокие знания, при ответе не допускает грубых ошибок или противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами, в решении задач могут содержаться грубые ошибки. Для получения правильного ответа требуются уточняющие вопросы.
Неудовлетворительно	Не выполнены обязательные **условия подготовки студента к промежуточной аттестации**, предусмотренные программой дисциплины, ИЛИ студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература

1. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2023. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/519714

2. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2023. — 218 с. — (Высшее образование). — ISBN 978-5-534-00515-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/512113

3. Кириченко, А. А., Дубовик, Е.В. Laravel для веб-разработчиков. Практическое руководство по созданию профессиональных сайтов – СПб.: Наука и техника, 2021.

4. Стаффер, Мэтт. Laravel. Полное руководство. 2-е изд. – СПб.: Питер, 2020 – 512 с..: ил. – (Серия «Бестселлеры O’Reilly»)

7.2. Дополнительная литература

1. Сысолетин, Е. Г. Разработка интернет-приложений : учебное пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под научной редакцией Л. Г. Доросинского. — Москва : Издательство Юрайт, 2022. — 90 с. — (Высшее образование). — ISBN 978-5-9916-9975-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/492224

2. Дронов, В.А. Laravel 8. Быстрая разработка веб-сайтов на PHP — Москва : Издательство BHV, 2021. — 688 с.

3. Маркин, А. В. Программирование на SQL в 2 ч. Часть 1 : учебник и практикум для вузов / А. В. Маркин. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 429 с. — (Высшее образование). — ISBN 978-5-534-15817-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/509818

8. Материально-техническое обеспечение дисциплины

8.1 Требования к оборудованию и помещению для занятий

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

8.2 Требования к программному обеспечению

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

1. Visual Studio Code.
2. OpenServer
3. MySQL
4. PHP
5. Веб-браузер.
6. Docker
7. Laravel
8. Composer
9. Git

9. Методические рекомендации для самостоятельной работы студентов

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, лабораторные работы.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста в области Веб-технологий.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на экзамене в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;

- оформление материала в соответствии с требованиями.

10. Методические рекомендации для преподавателя

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

Приложение 1

Структура и содержание дисциплины «Серверная веб-разработка»
Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки Веб-технологии (Бакалавр)

№	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов					Формы аттестации	
				Л	П / С	Лаб	СРС	КСР	К. Р.	К.П.	РГР	Реферат	К/р	Э	З
1	**Раздел 1. Ознакомление. Настройка окружения**	3	1-2	**8**			**16**								
1.1	Тема 1. Настройка окружения	3	3-4	8			16								
2	**Раздел 2. Основы работы с фреймворком**	3	5-8	**0**		**20**	**30**								
2.1	Тема 1. Маршрутизация	3	9-10			4	6								
2.2	Тема 2. Шаблонизатор Blade	3	11-13			4	6								
2.3	Тема 3. Контроллеры, экшены	3	14-15			4	6								
2.4	Тема 4. Обработка запроса (Request, валидация). Формирование ответа (Response, json).	3				4	6								
2.5	Тема 5. Работа с формами (CSRF).	3				4	6								
3	**Раздел 3. Работа над сайтом**	3		**0**		**44**	**60**								
3.1	Тема 1. Работа с БД. ORM-система Eloquent.	3				8	10								

3.2	Тема 2. Аутентификация и авторизация.	3				8	10								
3.3	Тема 3. Рассылка email. Broadcasting, events, notifications.	3				8	10								
3.4	Тема 4. Кэширование данных, очереди.	3				6	10								
3.5	Тема 5. Планировщик задач.	3				6	10								
3.6	Тема 6. Backend API.	3				8	10								
	Форма аттестации	3	**16-17**							1				**Э**	
	Всего часов по дисциплине			**8**		**64**	**108**								

Приложение 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Направление подготовки 09.03.01 Информатика и вычислительная техника
профиль подготовки «Веб-технологии»
Форма обучения: очная

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

Серверная веб-разработка

Состав:

1. **Показатель уровня сформированности компетенций.**
2. **Перечень оценочных средств.**
3. **Вопросы для экзамена.**
4. **Типовая программа экзамена.**

Москва, 2022 год

1. ПОКАЗАТЕЛЬ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

«Серверная веб-разработка»					
ФГОС ВО 09.03.01 «Информатика и вычислительная техника» профиль подготовки «**Веб-технологии»**					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **общепрофессиональные компетенции**:					
Компетенции		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства****	**Степени уровней освоения компетенций**
Индекс	**Индекс**				
ПК-1	Способен разрабатывать требования и проектировать программное обеспечение	ПК-1.1. Знать: методологии и технологии проектирования и использования баз данных; методы и средства проектирования баз данных. ПК-1.2. Уметь: использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов.	Лабораторные работы, Самостоятельная работа, индивидуальные консультации	УО П Экзамен	**БАЗОВЫЙ УРОВЕНЬ:** способность выполнять полученное задание, применяя полученные знание и умения на практике, владеть соответствующими индикаторами компетенции при выполнении задания. **ПРОДВИНУТЫЙ УРОВЕНЬ:** способность выполнять полученное задание и решать самостоятельно сформированные задачи, применяя полученные знание и умения на практике. Уверенно владеть соответствующими индикаторами компетенции при выполнении задания, комбинировать их между собой и с индикаторами других компетенций для достижения проектных результатов.

		ПК-1.3. Владеть: современным инструментарием и средами разработки программного кода.			
ПК-2	Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	ПК-2.2. Уметь: исполнять ручные тесты. ПК-2.3. Владеть: средствами автоматизации проектирования ПО.			
ПК-4	Способен выполнять работы по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-4.2. Уметь: разрабатывать справочники по интерфейсам прикладного программирования; разрабатывать руководство по языку программирования.			
ПК-5	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов	ПК-5.1. Знать: языки веб-разработки. ПК-5.3. Владеть: языками веб-разработки.			

**- Сокращения форм оценочных средств см. в приложении 2 к РП.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Устный опрос / собеседование, (УО)	Средство контроля, организованное как презентация обучающимся результатов выполнения Курсового проекта с демонстрацией наглядных материалов и ответов на вопросы педагогических работников (работника) на тему доклада, теме, проблеме и т.п.	Контрольные вопросы
2	Проект (П)	Конечный продукт, получаемый в результате планирования и выполнения комплекса учебных и исследовательских заданий. Позволяет оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских навыков, навыков практического и творческого мышления. Может выполняться в индивидуальном порядке или группой обучающихся.	Типовая программа экзамена

3. ВОПРОСЫ ДЛЯ ЭКЗАМЕНА

1. Что такое модель в Laravel?
2. Зачем нужен Artisan?
3. В чем преимущество Laravel перед созданием проекта с нуля на чистом PHP
4. Какую архитектуру реализует Django?
5. За что отвечает routes/web.php?
6. Когда нужно делать миграцию?
7. Какое преимущество у записи вида " Обратная связь" перед " Обратная связь" ?
8. Какое отношение описывает следующая ситуация: "Каждая статья соответствует одному разделу. А один раздел модет иметь много статей"
9. Какое отношение описыает следующая ситуация: "Каждый пользователь имеет много ролей. А каждая роль имеется у многих пользователей"
10. Что делают посредники?
11. Вам потребовалось изменить СУБД с MySQL на SQLite. Что для этого нужно сделать в Laravel?
12. Для чего нужны наполнитель (seeder)?
13. Зачем нужна директива @csrf
14. Чем планировщик заданий в Laravel лучше стандартного планировщика cron?
15. Что из этого списка является необходимым для работы Laravel?

4. ТИПОВАЯ ПРОГРАММА ЭКЗАМЕНА

Экзамен проводится по билетам. Билет включает в себя два практических задания и один теоретический вопрос, позволяющих комплексно оценить знания, умения и практические навыки по изученной

дисциплине. Время выполнения задания, без учета времени его проверки преподавателем 40 минут.

ВРЕМЯ ВЫПОЛНЕНИЯ ЗАДАНИЯ

40 минут.

ЗАДАНИЕ ЭКЗАМЕНА (Пример экзаменационного билета)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Экзамен по дисциплине: Серверная веб-разработка

Билет №7.

Базовое задание

Storage of things

Разработать веб-приложение для организации хранения вещей. Суть приложения в создании вещей и мест для их хранения. Хозяин вещи может передавать ее в пользование другим зарегистрированным пользователям в данном приложении.

Реализовать регистрацию новых пользователей и аутентификацию.

Создать три таблицы в БД: thing (для хранения вещей), place (место хранения), use (связывает пользователя с определенным количеством вещей).

Связать таблицы отношениями. Реализовать CRUD для вещей и мест хранения, включая создание преставлений для пользователя.

Таблица 1 thing: id, name, description, wrnt (гарантия/срок годности), master (создатель вещи, ее хозяин).

Таблица 2 place: id, name, description, repair (флаг специального места пребывания: ремонт, мойка), work (находится в работе).

Таблица 3 use: thing_id, place_id, user_id, amount (количество)

Дополнительное задание

Добавить в БД таблицу с правами доступа. Используя автозаполнение Laravel, добавить роль администратора. Все CRUD методы для работы с местом хранения доступны администратору. Использовать политики (Policy) Laravel для проверки авторизации пользователя на выполнение определенного действия.

Вопрос

Описать жизненный цикл запроса.

УСЛОВИЯ ВЫПОЛНЕНИЯ РАБОТЫ

В качестве ответа представьте ссылку на git-репозиторий.