Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 23.05.2024 10:24:59
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет «Информационных технологий»

УТВЕРЖДАЮ
Декан факультета
«Информационные технологии»
_____ / Д.Г.Демидов /
«15» февраля 2024г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Серверная веб-разработка

Направление подготовки/специальность

09.03.01 Информатика и вычислительная техника

Профиль/специализация

Веб-технологии

Квалификация

Бакалавр

Формы обучения

Очная, заочная

Москва, 2024 г.

Разработчик(и):



Преподаватель / Володина О.В. /
степень, звание, должность

Согласовано:
Заведующий кафедрой «Инфокогнитивные технологии»,



к.т.н., доцент / Е.А.Пухова /

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 6
3 Структура и содержание дисциплины 6
3.1 Виды учебной работы и трудоемкость 6
3.2 Тематический план изучения дисциплины 7
3.3 Содержание дисциплины 8
3.4 Тематика семинарских/практических и лабораторных занятий 10
4 Учебно-методическое и информационное обеспечение 11
4.1 Нормативные документы и ГОСТы 11
4.2 Основная литература 11
4.3 Дополнительная литература 12
4.4 Электронные образовательные ресурсы 12
4.5 Лицензионное и свободно распространяемое программное обеспечение 12
4.6 Современные профессиональные базы данных и информационные справочные системы 12
5 Материально-техническое обеспечение 12
6 Методические рекомендации 12
6.1 Методические рекомендации для преподавателя по организации обучения 12
6.2 Методические указания для обучающихся по освоению дисциплины 13
7 Фонд оценочных средств 13
7.1 Методы контроля и оценивания результатов обучения 13
7.2 Шкала и критерии оценивания результатов обучения 14
7.3 Оценочные средства 15

1 Цели, задачи и планируемые результаты обучения по дисциплине

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.

Обучение по дисциплине «Серверная веб-разработка» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИУК-1.1. Анализирует задачу, выделяя ее базовые составляющие ИУК-1.2. Осуществляет поиск, критически оценивает, обобщает, систематизирует и ранжирует информацию, требуемую для решения поставленной задачи ИУК-1.3. Рассматривает и предлагает рациональные варианты решения поставленной задачи, используя системный подход, критически оценивает их достоинства и недостатки
ОПК-7. Способен участвовать в настройке и наладке программно-аппаратных комплексов	ИОПК-7.1. Знает основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ИОПК-7.2. Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ИОПК-7.3. Владеет языком программирования, методами отладки и тестирования работоспособности программы

ПК-3. Способен разрабатывать требования и проектировать программное обеспечение	**ИПК-3.1. Знает** возможности существующей программно-технической архитектуры; возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования; методологии и технологии проектирования и использования баз данных; языки формализации функциональных спецификаций; методы и приемы формализации задач; методы и средства проектирования программного обеспечения; методы и средства проектирования программных интерфейсов; методы и средства проектирования баз данных; принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; методы и средства проектирования программного обеспечения и баз данных; методы и средства проектирования программных интерфейсов. **ИПК-3.2.** Умеет проводить анализ исполнения требований; вырабатывать варианты реализации требований; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; выбирать средства реализации требований к программному обеспечению; вырабатывать варианты реализации программного обеспечения; проводить оценку и обоснование рекомендуемых решений; осуществлять коммуникации с заинтересованными сторонами; использовать существующие типовые решения и шаблоны проектирования программного обеспечения; применять методы и средства проектирования программного обеспечения, структур

	данных, баз данных, программных интерфейсов; осуществлять коммуникации с заинтересованными сторонами. **ИПК-3.3. Владеет** современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений.

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

- Администрирование серверов;
- Основы программирования;
- Основы веб-технологий;
- Веб-разработка;
- Базы данных;
- Основы серверной веб-разработки.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зачетные единицы (180 часа).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
				3
1	**Аудиторные занятия**	**72**		72
	В том числе:			
1.1	Лекции	18		18
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	54		54
2	**Самостоятельная работа**	**108**		108
	В том числе:			
2.1	Курсовой проект	КП		КП
2.2	…			
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/экзамен	**экзамен**		экзамен
	Итого:	**180**		180

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	**Раздел 1. Ознакомление. Настройка окружения**	**31**	**8**	**0**	**0**	**0**	**23**
1.1	Тема 1. Настройка окружения	31	8				23
2	**Раздел 2. Основы работы с фреймворком**	**35**	**10**	**0**	**10**	**0**	**25**
2.1	Тема 1. Маршрутизация	9	2		2		5
2.2	Тема 2. Шаблонизатор Blade	9	2		2		5
2.3	Тема 3. Контроллеры, экшены	9	2		2		5
2.4	Тема 4. Обработка запроса (Request, валидация). Формирование ответа (Response, json).	9	2		2		5
2.5	Тема 5. Работа с формами (CSRF).	9	2		2		5
3	**Раздел 3. Работа над сайтом**	**104**	**0**	**0**	**44**	**0**	**60**
3.1	Тема 1. Работа с БД. ORM-система Eloquent.	19			8		11
3.2	Тема 2. Аутентификация и авторизация.	18			8		10
3.3	Тема 3. Рассылка email. Broadcasting, events, notifications.	18			8		10
3.4	Тема 4. Кэширование данных, очереди.	16			6		10
3.5	Тема 5. Планировщик задач.	16			6		10
3.6	Тема 6. Backend API.	19			8		11
Итого		**144**	**18**	**0**	**54**	**0**	**72**

3.3 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Заочная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
				4
1	**Аудиторные занятия**	**16**		72
	В том числе:			
1.1	Лекции	4		18
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	12		54
2	**Самостоятельная работа**	**164**		164
	В том числе:			
2.1	Курсовой проект			КП
2.2	…			
3	**Промежуточная аттестация**			
	Зачет/диф.зачет/экзамен	**экзамен**		экзамен
	Итого:	**180**		180

3.4 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Заочная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	**Раздел 1. Ознакомление. Настройка окружения**	**31**	**2**	**0**	**0**	**0**	**23**
1.1	Тема 1. Настройка окружения	15	2				13
2	**Раздел 2. Основы работы с фреймворком**	**35**	**2**	**0**	**6**	**0**	**25**
2.1	Тема 1. Маршрутизация	17	2		2		13
2.2	Тема 2. Шаблонизатор Blade	15			2		13
2.3	Тема 3. Контроллеры, экшены	15			2		13
2.4	Тема 4. Обработка запроса (Request, валидация). Формирование ответа (Response, json).	13					13
2.5	Тема 5. Работа с формами (CSRF).	13					13
3	**Раздел 3. Работа над сайтом**	**104**	**0**	**0**	**6**	**0**	**60**
3.1	Тема 1. Работа с БД. ORM-система Eloquent.	15			2		13
3.2	Тема 2. Аутентификация и авторизация.	15			2		13
3.3	Тема 3. Рассылка email. Broadcasting, events, notifications.	15			2		13

3.4	Тема 4. Кэширование данных, очереди.	13					13
3.5	Тема 5. Планировщик задач.	13					13
3.6	Тема 6. Backend API.	21					21
Итого		**180**	**4**	**0**	**12**	**0**	**164**

3.5 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).

3.6 Тематика семинарских/практических и лабораторных занятий

3.4.1 Семинарские/практические занятия

3.4.2 Лабораторные занятия

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»

4 Учебно-методическое и информационное обеспечение

4.1 Нормативные документы и ГОСТы

1. Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);

2. Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденный Приказом Министерства образования и науки РФ от 19 сентября 2017 г. N 929 "Об утверждении федерального… Редакция с изменениями N 1456 от 26.11.2020

3. Приказ Министерства образования и науки РФ от 05 апреля 2017 г. № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры;

4. Порядок проведения государственной итоговой аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры, утвержденный приказом Минобрнауки России от 29 июня 2015 г. № 636;

5. Положение о практической подготовке обучающихся, утвержденное приказом Министерства науки и высшего образования Российской Федерации и Министерства просвещения Российской Федерации от 5 августа 2020 г. № 885/390;

6. Устав и локальные нормативные акты Московского политеха

Области профессиональной деятельности и сферы профессиональной деятельности, в которых выпускники, освоившие программу бакалавриата (далее - выпускники), могут осуществлять профессиональную деятельность:

06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки, внедрения и эксплуатации средств вычислительной техники и информационных систем, управления их жизненным циклом)

Выпускники могут осуществлять профессиональную деятельность в других областях и (или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника, предъявляемым соответствующими профессиональными стандартами

4.2 Основная литература

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

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

4.3 Дополнительная литература

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

2. 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

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

4.4 Электронные образовательные ресурсы

1. Курс в LMS «Серверная веб-разработка, часть 1, курс 1, семестр 2» https://lms.mospolytech.ru/course/view.php?id=9947
2. Курс в LMS «Серверная веб-разработка, часть 2, курс 2, семестр 3» https://lms.mospolytech.ru/local/crw/course.php?id=12516.

4.5 Лицензионное и свободно распространяемое программное обеспечение

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

4.6 Современные профессиональные базы данных и информационные справочные системы

5 Материально-техническое обеспечение

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.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

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.

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

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи с 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, 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.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

Вводная лекция включает представление и описание итоговой работы, необходимой для оценивания уровня знаний по дисциплине. Студенты, чей уровень знаний достаточен для выполнения представленной работы самостоятельно, могут выполнить итоговую работу и защитить ее в течение первого месяца обучения. Защита состоится очно в аудитории и включает демонстрацию программного кода, написанного с использованием фреймворка Laravel и ответы на вопросы по архитектуре фреймворка. По результатам защиты студент получает итоговую экзаменационную оценку.

Лабораторные работы и дополнительные задания необходимо защищать не позднее недели следующей за неделей проведения лабораторной работы в аудитории.

После рассмотрения пятой темы третьего раздела дисциплины студенты очно в аудитории защищают выполненный проект в день, назначенный преподавателем. Защита включает демонстрацию программного кода, написанного с использованием фреймворка Laravel. Дата защиты работы объявляется за неделю. Студенты защитившие проект в объявленный день, получают итоговую экзаменационную оценку.

По завершении изучения дисциплины студенты защищают backend API. Защита состоится очно в аудитории и включает демонстрацию программного кода, написанного с

использованием фреймворка Laravel, и демонстрацию функционирования с помощью удобной API платформы. Студенты защитившие backend API в объявленный день, получают итоговую экзаменационную оценку.

7.2 Шкала и критерии оценивания результатов обучения

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (обязательные условия подготовки студента), 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 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 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 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 студентом допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями.

7.3 Оценочные средства

7.3.1 Текущий контроль

Вопросы итогового теста

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?

7.3.2 Промежуточная аттестация

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.

КРИТЕРИИ ОЦЕНКИ ЗАДАНИЯ

Оценка "Отлично" - предполагает выполнение всех заданий и успешно выполненные лабораторные работы с домашними заданиями.

Оценка "Хорошо" - предполагает выполнение всех заданий и успешно выполненные лабораторные работы.

Оценка "Удовлетворительно" - предполагает выполнение базового задания, ответ на теоретический вопрос и выполнение минимум 7 лабораторных работ.

Набранный уровень достижения и соответствующая им оценка имеет рекомендательный характер – экзаменатор имеет право скорректировать оценку в ту или иную сторону.