

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 11.06.2026 11:50:14
Уникальный программный ключ:
e3a68f3eaa1e62674b54f4998099d3d6b6dcf836

Оценочные материалы для промежуточной аттестации по дисциплине

Технология разработки web-приложений

Код, направление подготовки	27.04.04 Управление в технических системах
Направленность (профиль)	Управление и информатика в технических системах
Форма обучения	Очная
Кафедра-разработчик	Автоматики и компьютерных систем
Выпускающая кафедра	Автоматики и компьютерных систем

Типовые задания для контрольной работы:

Задание 1

1. Для реализации архитектуры SPA необходимо дополнить проект на базе Codeigneter функциями, реализующими REST API, включая аутентификацию. Промышленным стандартом средств аутентификации является протокол OAuth 2.0. Для его поддержки необходимо установить библиотеку OAuth2 Server Library for PHP (by Brent Shaffer).
2. В папке проекта выполнить команду `composer require bshaffer/oauth2-server-php "^1.10"`. После этого в файл `composer.json` будет добавлена соответствующая зависимость, а необходимые файлы появятся в папке `vendor/bshaffer`.
3. Создать миграцию при помощи команды `php spark make:migration AddBshafferOAuthTables`
4. Выполнить команду `php spark migrate` для запуска миграции.
5. В папке `Database/Seeds` создать файл, например `Bshafferoauth.php` и добавить в него следующий код для создания в БД записи о тестовом клиенте.
6. При помощи команды `php spark db:seed` "засеять" базу данных.
7. Для переопределения класса PDO в папке `app/services` создать файл `MyPdo.php`. В классе `MyPdo` переопределены методы шифрования и проверки пароля, которые, в свою очередь, используют соответствующие методы класса `IonAuthModel`.
8. В соответствии с приведенными выше инструкциями реализовать в проекте REST API для аутентификации клиента и получения данных из БД.
9. В качестве ответа на задание прикрепить ссылку на приложение на сервере Heroku или другом хостинге, ссылку на репозиторий и отчет со скриншотами выполненных к API запросов.

Задание 2

1. Установить менеджер пакетов npm: `sudo apt install npm`
2. Установить процессор команд Vue CLI: `sudo npm install -g @vue/cli`
3. На сайте `bitbucket.org` или аналогичном создать git-репозиторий.
4. Установить среду разработки JetBrains WebStorm: `sudo snap install webstorm --classic`
5. Скопировать URL репозитория с сайта BitBucket.
6. Вставить в форму создания проекта WebStorm и нажать Clone.
7. Для установки VueJS папку проекта нужно в терминале перейти в родительскую папку (`WebstormProjects`) и выполнить следующую команду (имя проекта должно совпадать с именем папки, то есть с именем репозитория!):

8. Открыть инструменты разработчика в браузере (Ctrl-Shift-I), найти запросы на аутентификацию и на получение информации о пользователе.
9. В отчет поместить скриншоты и структуры данных Json, полученные приложением от сервера.

Задание 3

1. Для установки библиотеки компонентов PrimeVue в корневой папке проекта выполнить команду: `npm install primevue@^3.9.1 --save`
2. Для установки библиотеки иконок: `npm install primeicons --save`
3. В файле main.js необходимо выполнить импорт библиотеки: `import PrimeVue from 'primevue/config'`;
4. Добавить импорт стилей: `import 'primevue/resources/themes/bootstrap4-light-blue/theme.css'`; `import 'primeicons/primeicons.css'`; `import 'primeflex/primeflex.css'`; `import 'primevue/resources/primevue.min.css'`
5. Добавить подключение библиотеки: `myApp.use(PrimeVue)`;
6. Доработать Vue3 приложение, добавив функции постраничного отображения записей из какой-либо таблицы БД с использованием библиотеки компонентов PrimeVue.
7. В качестве ответа на задание прикрепить ссылку на репозиторий и отчет со скриншотами работы приложения.
8. Доработать Vue3 приложение, добавив функции постраничного отображения записей из какой-либо таблицы БД.
9. В качестве ответа на задание прикрепить ссылку на репозиторий и отчет со скриншотами работы приложения и анализа запросов к API в браузере.

Типовые вопросы к экзамену:

1. Назовите основные принципы ООП.
2. Что такое наследование, инкапсуляция, абстракция, полиморфизм: приведите примеры (желательно из собственного опыта). От какого класса неявно наследуются все классы в .NET? Разрешено ли множественное наследование в C#?
3. Что такое рекурсия?
4. Что такое лямбда-выражение?
5. Что такое параллельное программирование (многопоточность) и его назначение? Какие классы используются?
6. Что такое JSON?
7. Как вы понимаете REST?
8. Расскажите о SPA concept.
9. Какие GoF-паттерны использовали?
10. Какая разница между GET и POST HTTP методами?
11. Какую проблему решает Docker? Каковы его плюсы и минусы?
12. Чем принципиально отличаются unit-тесты от интеграционных тестов?
13. Какая директива используется для создания двунаправленной привязки данных?