

Документ подписан электронной подписью  
Информация о владельце:  
ФИО: Косенок Сергей Михайлович  
Должность: ректор  
Дата подписания: 20.06.2024 11:24:56  
Уникальный программный ключ:  
e3a68f3eaa1a676746541d99819c47d661dct1836

Оценочные материалы для промежуточной аттестации по дисциплине

## Разработка web-приложений с архитектурой SPA

Код, направление	09.04.04 ПРОГРАММНАЯ ИНЖЕНЕРИЯ
подготовки	
Направленность (профиль)	Разработка и интеграция информационных систем и сервисов
Форма обучения	Очная
Кафедра-разработчик	Автоматики и компьютерных систем
Выпускающая кафедра	Автоматики и компьютерных систем

**Этап: проведение текущего контроля успеваемости по дисциплине (вопросы к лабораторным работам)**

### Раздел 1. Базовые технологии приложения SPA

#### Тема: Реализация REST API

##### *Лабораторная работа*

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.

#### Тема: Аутентификация пользователей в приложении на VueJS

##### *Лабораторная работа*

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.

### **Тема: Постраничный вывод данных в приложении на VueJS**

#### *Лабораторная работа*

1. Доработать Vue3 приложение, добавив функции постраничного отображения записей из какой-либо таблицы БД.
2. В качестве ответа на задание прикрепить ссылку на репозиторий и отчет со скриншотами работы приложения и анализа запросов к API в браузере.

### **Тема: Работа с компонентами библиотеки PrimeVue**

#### *Лабораторная работа*

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.

## **Раздел 2. Архитектура и интеграции SPA приложения**

### **Тема: Разработка CRUD-приложения на VueJS**

#### *Лабораторная работа*

1. В API-контроллер необходимо добавить метод для создания записей.
2. Проверить работу метода по созданию записей при помощи Postman (в отчете привести скриншот с результатом запроса). При этом необходимо указать действующий токен.
3. В раздел Body добавить данные, нажать кнопку Send. В случае успешного выполнения запроса в ответе будет содержаться код 201 Rating created successfully.
4. Проверить работу метода API по валидации данных формы, введя некорректные значения form-data (в отчете привести скриншот с результатом).
5. В объект `vueX` (файл `store.js`) необходимо добавить переменную `createRatingDialogVisible` для управления видимостью диалога создания рейтинга, метод `setCreateRatingDialogVisible` для установки значения этой переменной, переменную `Validation` для хранения ошибок валидации, метод `setValidation` для установки ее значений, а также методы `createRating` и `deleteRating`.
6. В компоненте отображения данных `Rating.vue` можно использовать для отображения элемента данных `Card` (можно оставить `Panel`). Необходимо добавить кнопки для удаления и редактирования (пока не реализовано) элементов. Также добавить в компонент диалог подтверждения удаления `ConfirmDialog`. Добавленные строки выделены.
7. В соответствии с приведенным выше кодом доработать Vue3 приложение, добавив функции создания и удаления записей в какой-либо таблице БД.
8. В качестве ответа на задание прикрепить ссылку на репозиторий и отчет со скриншотами работы приложения.

## **Тема: Размещение VueJS-приложения на хостинге Heroku**

### *Лабораторная работа*

1. Для работы frontend-приложения на внешнем сервере backend-приложение также должно быть доступно в глобальной сети. Поскольку API реализован как часть MVC-приложения на Codeigniter, то требуется просто обновить приложение, развернутое на хостинге Heroku.
2. Запустить PHP Storm.
3. В меню Git/Manage remotes проверить подключение к репозиторию на Heroku.
4. При помощи меню Git/Commit выполнить коммит проекта и синхронизацию с репозиторием (Push). При выполнении Push выбрать репозиторий heroku.
5. Проверить работу MVC-приложения, в том числе функции добавления, удаления и редактирования записей.
6. В соответствии с приведенными выше инструкциями развернуть и отладить VueJS-приложение на хостинге Heroku (или любом другом хостинге). В качестве ответа на задание приложить отчет, содержащий: ссылку на репозиторий Vue-JS приложения на Bitbucket; скриншот работающего приложения; ссылку на приложение; логин и пароль администратора для входа через web-интерфейс.

## **Тема: Работа с AWS S3-совместимым хранилищем в приложении VueJS**

### *Практическая работа*

1. Доступ для чтения из хранилища открыт, но просмотр списка файлов хранилища для пользователя student недоступен, то есть файл доступен только если известна прямая ссылка и уникальное имя файла. При загрузке необходимо обеспечить уникальность имени файла.
2. Для тестирования запустить утилиту Postman и убедиться в работоспособности метода. В отчете привести скриншот с результатом выполнения запроса.
3. В соответствии с приведенным выше кодом доработать Vue3 приложение, добавив функции загрузки файла в хранилище при создании записей в какой-либо таблице БД.
4. В качестве ответа на задание прикрепить ссылку на репозиторий и отчет со скриншотами работы приложения, а также ссылку на VueJS-приложение, размещенное на хостинге Heroku или другом хостинге.

## **Тема: Аутентификация через сервис Google в приложении VueJS**

### *Практическая работа*

1. Необходимо выбрать одну из сущностей в БД согласно своему варианту задания, с которой может быть ассоциировано одно или несколько изображений и реализовать для нее загрузку изображений в формах создания, редактирования и вывод изображений в интерфейсе веб-приложения.
2. Для получения доступа (логина и пароля) к хранилищу S3 необходимо направить запрос преподавателю (в свободной форме, с использованием электронной почты или другого вида сообщений). В запросе сообщить идентификатор, состоящий из номера группы и инициалов в формате 60971xxx (будет являться логином). В ответе преподаватель сообщит пароль.
3. В качестве ответа на задание прикрепить ссылку на приложение на сервере Heroku или другом хостинге и ссылку на репозиторий.

## **Разработка SPA-приложения на VueJS (Контрольная работа)**

1. С помощью приложения PhpMyAdmin (<http://60971xxx.student.surgu.ru/phpmyadmin/>) переименовать БД проекта (чтобы потом создать БД с таким же именем).
2. В папке app/Database/Migrations при помощи команды `php spark migrate:make` создать два файла (две миграции) для таблиц библиотеки IonAuth (содержимое файла для IonAuth можно взять из описания выше без изменений) и для своей БД (разработать по примеру).
3. Запустить миграции при помощи команды `php spark migrate`. Сделать скриншот с результатами успешного выполнения команды Иконка скриншот

4. В папке `app/Database/Seeds` создать два файла для первичного наполнения таблиц библиотеки `IonAuth` (содержимое файла для `IonAuth` можно взять из описания выше без изменений) и для своей БД (разработать по примеру).
5. "Засеять" таблицы данными при помощи команды `php spark db:seed <имя_файла>`. Сделать скриншот с результатами успешного выполнения команды
6. Проверить корректность работы приложения с вновь созданной БД. Сделать скриншот с выводом данных из БД.
7. Синхронизировать проект с удаленным репозиторием. В тексте задания разместить ссылку на репозиторий.

### **Этап: проведение промежуточной аттестации по дисциплине (ЭКЗАМЕН)**

1. Как работает наследование?
2. Что такое замыкание и зачем оно нужно?
3. Как искать элементы в DOM-дереве?
4. Как работает Event loop?
5. Что такое this-объект?
6. На странице отображается больше тысячи позиций, и страница начинает тормозить. Как это лучше исправить?
7. Что такое типы данных и как их сравнивать? Чем отличается `==` от `===` ? ...
8. Как работают методы . ...
9. Что такое Promise, зачем нужен `async / await` ?
10. Зачем нужен Vue? Почему нельзя обойтись без него?
11. Зачем нужны различные методологии и подходы при работе с CSS?
12. Архитектура и структура баз данных (БД)
13. Простые запросы, `join`ы`, агрегаты
14. Базовые команды в SQL и встроенные аналитические функции
15. Импорт и экспорт данных посредством SQL и ETL программ
16. Принципы работы с различными БД
17. Основные библиотеки для подключения к БД из Python
18. Функции SQL и их аналоги в pandas
19. Консоль (знакомство, основные операторы, утилита `psql`)
20. Архитектура и проектирование
21. Нормализация