Бюджетное профессиональное образовательное учреждение Омской области «Омский колледж отраслевых технологий строительства и транспорта»

Рекомендуемая продолжительность выполнения пунктов задания демонстрационного экзамена

**Методические указания**

**Руководство по прохождению демонстрационного экзамена КОД 1.4-2022-2024**

**Практическая часть**

Ч е р н о в и к

Омск 2023

**Структура Веб-сайта**

LOGO

Панель Администратора

Dynamic Количество решенных ЗАЯВОК on change play

Ajax СЧЕТ 2 notif.mp3

10 АПРЕЛЯ

(заявка 2)

**Статус «Решена»**

**РЕМОНТ ДОРОГ**

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст ТекстТекст Текст

15 АПРЕЛЯ

(заявка 2 - исполнение)

**РЕМОНТ ДОРОГ**

IMG (ПОСЛЕ/ДО)

Применить масштабирование при наведении курсора с заменой IMG

При необходимости добавить futer

3 АПРЕЛЯ

(заявка 1)

**Статус «Решена»**

**УБОРКА МУСОРА**

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст ТекстТекст Текст

4 АПРЕЛЯ

(заявка 2 - исполнение)

**УБОРКА МУСОРА**

IMG (ПОСЛЕ/ДО)

Применить масштабирование при наведении курсора с заменой IMG

Login

Register

Выход

Личный кабинет

Рис. 1. Главная страница. Категории ссылок «Login» (Авторизация), «Register» (Регистрация), «Личный кабинет», «Панель Администратора» (скрыта до процедуры авторизации)

**АВТОРИЗАЦИЯ (LOGIN)**

Login

Пароль

Войти

Рис. 2. Страница «Авторизация»

**РЕГИСТРАЦИЯ (REGISTER)**

ФИО

Login

E-mail

Пароль

Подтвердить пароль

Согласие на обработку персональных данных

РЕГИСТРАЦИЯ

Рис. 2. Страница «Регистрация»

ЛИЧНЫЙ КАБИНЕТ

Все

Решенные

Отклоненные

Показать только

ВЫХОД

ЗАЯВКА 1

**Вариант 2.**

Время Описание

Категория

Статус

**ПРОСМОТРЕТЬ**

Решена

ЗАЯВКА 1

**Вариант 3.**

Время Описание

Категория

Статус

**ПРОСМОТРЕТЬ**

Отклонена

ЗАЯВКА 2

**Вариант 1**

Время Описание

Категория

Статус

**ПРОСМОТРЕТЬ**

**УДАЛИТЬ**

ЗАЯВКА 1. **Вариант 1**

Время Описание

Категория

Статус

**ПРОСМОТРЕТЬ**

**УДАЛИТЬ**

УДАЛИТЬ?

ДА

НЕТ

ЗАЯВКА 2

**Вариант 2**

Время Описание

Категория

Статус

**ПРОСМОТРЕТЬ**

Решена

ЗАЯВКА 2

**Вариант 3.**

Время Описание

Категория

Статус

**ПРОСМОТРЕТЬ**

Отклонена

ВЫЙТИ?

ДА

НЕТ

Создать заявку

Рис. 4. Страница «Личный кабинет» Авторизованного пользователя. Категории ссылок: «Создать заявку», «Просмотреть заявку». «Посмотреть», «удалить», выполняется в виде кнопок

Фото, демонстрирующее проблему в одном из форматов (jpg, jpeg, png, bmp) не должно превышать 10Мб

Имя.\*

ввод

Название

ввод

Описание

Категория

Категория 1. Ремонт дорог

Категория 2. Уборка мусора

Категория 3. Ремонт детских площадок

Категория 4. Ремонт зданий

Категория 5. Ремонт сооружений

Загрузить файл

Создать Заявку

**СОЗДАТЬ ЗАЯВКУ**

Заявка Зарегистрирована

Загружен

ОШИБКА

Заполнить все поля

Рис. 5. Страница «Создать Заявку». При невыполнении хотя бы одного из Пунктов, заявка не сохраняется, выводится сообщения об ошибке

Фото, демонстрирующее проблему в одном из форматов (jpg, jpeg, png, bmp) не должно превышать 10Мб

**ПРОСМОТРЕТЬ ЗАЯВКУ**

Название

Описание

Категория 1. Ремонт дорог

Категория 2. Уборка мусора

Категория 2. Ремонт детских площадок

Категория 2. Ремонт зданий

Категория 2. Ремонт сооружений

Изменить фото

Загруженне фото

IMG (ДО)

Заявитель: ФИО

Заявитель: E-mail

ЗАГРУЗИТЬ

Создана:

Изменена:

Изменить

Изменить

Категория

Рис. 6. Страница «Изменить Заявку Авторизованным пользователем». Обращаем внимание, изменение заявки возможно ДО присвоения Статуса

LOGO

Панель Администратора

Dynamic Количество решенных ЗАЯВОК on change play

Ajax СЧЕТ notif.mp3

10 АПРЕЛЯ

(заявка 2, импорт)

**РЕМОНТ ДОРОГ**

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст ТекстТекст Текст

15 АПРЕЛЯ

(заявка 2 исполнение)

**РЕМОНТ ДОРОГ**

IMG (ПОСЛЕ/ДО)

Применить масштабирование при наведении курсора с заменой IMG

При необходимости добавить futer

3 АПРЕЛЯ

(заявка 1, импорт)

**УБОРКА МУСОРА**

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст ТекстТекст Текст

4 АПРЕЛЯ

(заявка 2 исполнение)

**УБОРКА МУСОРА**

IMG (ПОСЛЕ/ДО)

Применить масштабирование при наведении курсора с заменой IMG

Login

Register

Выход

Личный кабинет

ВЫЙТИ?

ДА

НЕТ

Рис. 7. Страница «Выход». Здесь IMG (ДО) - фотография загруженная заявителем; IMG (ПОСЛЕ) - фотография загруженная Администратором и является доказательство решения проблемы

Время Описание

Категория

Статус

Загруженное фото IMG(ДО)

ИЗМЕНИТЬ

УДАЛИТЬ

Время Описание

Категория

Статус

Загруженное фото IMG(ДО)

ИЗМЕНИТЬ

УДАЛИТЬ

**ПАНЕЛЬ АДМИНИСТРАТОРА**

**ЗАЯВКИ**

Время Описание

Категория

Статус

Загруженное фото IMG (ДО)

ИЗМЕНИТЬ

УДАЛИТЬ

УДАЛИТЬ?

ДА

НЕТ

Время Описание

Категория

Статус

Загруженное фото IMG(ДО)

ИЗМЕНИТЬ

УДАЛИТЬ

ИЗМЕНИТЬ?

ДА

НЕТ

Рис. 8. Панель Администратора, «Заявки»

**ПАНЕЛЬ АДМИНИСТРАТОРА**

**ИЗМЕНИТЬ ЗАЯВКУ**

Название

Описание

Категория 1. Ремонт дорог

Категория 2. Уборка мусора

Категория 3. Ремонт детских площадок

Категория 4. Ремонт зданий

Категория 5. Ремонт сооружений

Статус

Решена

Отклонена

Загруженне фото

IMG (ДО)

Заявитель: ФИО

Заявитель: E-mail

Загруженное фото

IMG (ПОСЛЕ)

ЗАГРУЗИТЬ

Создана:

Изменена:

Ввести категорию

Изменить

Изменить

Категория

УДАЛИТЬ КАТЕГОРИЮ?

ДА

НЕТ

Рис. 7. Панель Администратора, «Заявки». Смена Статуса «Решена» на «Отклонена» невозможна

1. **Подготовительная часть**

*Указание:* Студент приходит на экзамен с документами в установленное время. Проходит инструктаж и приступает к выполнению задания демонстрационного экзамена.

1. Выполнить организационные условия, доложить о готовности к демонстрационному экзамену своему куратору (преподавателю) колледжа.
2. Сесть за рабочее место.
3. Запустить ПК.
4. Проверить наличие следующего установленного ПО на локальном ПК:

|  |
| --- |
| **Yii2**  **Laravel**  **Visual Studio Code**  **Open Server Panel**  **Zeal** |

В случае отсутствия установленного вышеперечисленного ПО доложить членам экзаменационной комиссии.

1. **Анализ задания демонстрационного экзамена**

*Указание:* Студент внимательно изучает задание демонстрационного экзамена, прорабатывает графический макет сайта, запускает приложения.

1. Ознакомиться с техническим заданием (выдается на экзамене).
2. Проработать архитектуру сайта (пример приведен на рисунках).
3. Запустить приложения:

|  |
| --- |
| Visual Studio Code  OpenServer  Zeal  Yii2/Laravel |

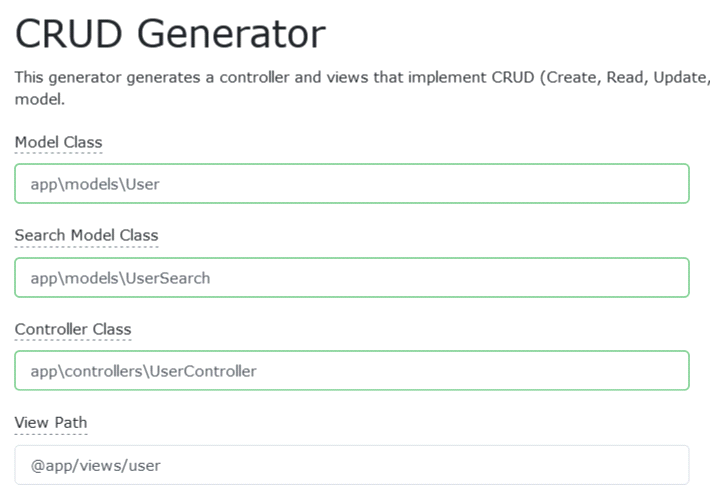
**3. Создание проекта на OpenServer, план работ**

*Указание:* студент создает папку проекта в OpenServer на локальном компьютере,и после, создает и настраивает базу данных.

1. Создать папку для проекта в domains (на локальном ПК).
2. Создать БД, присвоить название согласно технического задания.
3. Создание таблиц осуществляется путем поименования названий согласно технического задания (Пользователи, заявки и тд) согласно возможностям авторизированного пользователя.
4. Создание связи таблиц (в phpmyadmin в разделе «Связь» настроить связь между таблицами).
5. **Настройка фреймворка Yii2**

*Указание:* студент выбирает самостоятельно фреймворк Yii2/Laravel для выполнения задания. Предварительно, в настройках Open Server выбрать php не ниже версии 7.2.

1. Создать домен, указать папку домена basic/web/.
2. В папке config в файле web.php прописать cookieValidationKey, раскомментировать url менеджер для ЧПУ(числовое программное управление).
3. Создать базу данных в phpMyАdmin назвав БД согласно технического задания на демонстрационный экзамен.
4. В db.php прописать имя созданной базы данных, в соответствии с предыдущим пунктом.
5. Открыть инструмент «имя домена/gi»i, создать модели для таблиц базы данных через генератор, старый код user.php перед этим сохранить.
6. Создать контроллеры CRUD через генератор.



1. В папке views/layouts в файле main.php заменить кнопку «контакты» на «Регистрацию/Register» и указать ссылку /user/create на созданный контроллер (действие выполняется в случае требуемых заданием авторизации и регистрация пользователей в БД).
2. **Изменения в user.php и создание RegForm.php**

*Указания:* Студент настраивает файлы user.php и RegForm.php.

1. В объявление класса добавить implements \yii\web\IdentityInterface.
2. Скопировать методы реализации IdentityInterface из старого кода user.php.
3. В методе findIdentity прописать:

**return self::findOne($id)**

findIdentityByAccessToken содержимое тела метода убрать.

1. В findByUsername прописать:

**return self::find()->where(['login' => $username])->one();**

getAuthKey, validateAuthKey содержимое убрать

1. Делаем копию user.php в папке models, называем файл RegForm.php.
2. В файле RegForm.php переименовываем класс User на RegForm и пишем, что он является «extends User».
3. В файле RegForm.php функцию «tableName» удаляем.
4. В файле RegForm.php удаляем методы реализации интерфейса.
5. **Настройка формы регистрации и запросы к валидатору**

*Указание:* валидаторы в задании могут описывать персональные данные и данные контактов. В Yii2 валидаторы расположены в пространстве имен yii\validators.

1. По заданию, возможно потребуется добавить повтор пароля и согласие на обработку персональных данных. Для решения этой задачи в RegForm.php пропишем :

**public $passwordConfirm;**

**public $agree;**

1. В attributeLabels добавить новые поля, переименовать старые:

**'passwordConfirm' => 'Подтверждение пароля',**

**'agree' => 'Даю согласие на обработку данных',**

1. В файле UserController.php поменять модель с User на RegForm, в use сверху прописать:

**use app\models\RegForm**

1. Путь views/user/\_form.php – поправить форму, убрать строку администратора и добавить:

**<?= $form->field($model, 'passwordConfirm')->passwordInput(['maxlength' => true]) ?>**

**<?= $form->field($model, 'agree')->checkbox() ?>**

1. В rules добавить:

**'passwordConfirm', 'agree' и 'message' => 'Поле обязательное для заполнения'**

1. Добавляем валидатор «match», в соответствии с требованиями задания для внесения в БД ФИО (фамилия, имя, отчество) через и выражение:

**'/^[А-Яа-я\s\-]{5,}$/u'**

|  |  |  |
| --- | --- | --- |
| **Валидатор match:** проверяет, что входящее значение совпадает с указанным регулярным выражением | | |
| **Атрибут** | **Соответствие** | **Значение по умолчанию** |
| pattern | выражение, с которым должно совпадать входящее значение | Должно быть установлено (логин, фио и другие данные), иначе будет ошибка |
| not | инвертирует регулярное выражение | false |

1. Добавляем валидатор для логина как это показано выше, только условием будут латинские буквы. Логин должен быть уникальным, валидатор добавляем через «unique».

|  |  |  |
| --- | --- | --- |
| **Валидатор unique:** проверяет, что входящее значение совпадает с указанным регулярным выражением | | |
| **Атрибут** | **Соответствие** | **Значение по умолчанию** |
| targetClass | имя класса Active Record, который должен быть использован для проверки значения во входящих данных | будет использован класс модели, которая в данный момент проходит проверку |
| targetAttribute | имя атрибута в targetClass, который должен быть использован для проверки на уникальность входящего значени | будет использован атрибут, проверяемый в данный момент |
| filter | дополнительный фильтр, который можно присоединить к запросу в БД, чтобы использовать его при проверке значения на уникальность. Это может быть строка или массив |  |

Добавляем валидатор для ввода электонной почты, email.

|  |  |  |
| --- | --- | --- |
| **Валидатор email:** проверяет, что значение данных является корректным email-адресом | | |
| **Атрибут** | **Соответствие** | **Значение по умолчанию** |
| allowName | можно ли передавать в атрибут имя | false |
| checkDNS | существует ли доменное имя для введенного адреса | false |
| enableIDN | нужно ли учитывать IDN (многоязычные доменные имена) | false |

1. Добавляем валидатор «compare» для повтора ввода пароля.

|  |  |  |
| --- | --- | --- |
| **Валидатор compare:** валидатор сравнивает значение указанного атрибута с другим, чтобы удостовериться, что их отношение соответствует описанному в свойстве operator | | |
| **Атрибут** | **Соответствие** | **Значение по умолчанию** |
| compareAttribute | имя атрибута, с которым нужно сравнить значение. Например, если проверяющийся атрибут - password, то значение свойства по умолчанию будет password\_repeat | \_repeat |
| compareValue | постоянное значение, с которым будут сравниваться входящие данные | Password\_repeat |
| operator | оператор сравнения. Поддерживаются операторы:  ==: проверяет два значения на эквивалентность.  ===: проверяет два значения на эквивалентность.  !=: проверяет, что два значения не эквивалентны.  !==: проверяет, что два значения не эквивалентны.  >: проверяет, что валидируемое значение больше, чем то, с которым происходит сравнение.  >=: проверяет, что валидируемое значение больше или равно тому, с которым происходит сравнение.  <: проверяет, что валидируемое значение меньше, чем то, с которым происходит сравнение.  <=: проверяет, что валидируемое значение меньше или равно тому, с которым происходит сравнение. | == |
| type | Значения сравниваются побайтово. При сравнении чисел необходимо задать $type как 'number'. | string |

1. Добавляем валидатор для согласия с условиями «boolean» и «compare».

|  |  |  |
| --- | --- | --- |
| **Валидатор boolean:** проверяет, что второе значение является boolean | | |
| **Атрибут** | **Соответствие** | **Значение по умолчанию** |
| trueValue | значение, соответствующее true | 1 |
| falseValue | значение, соответствующее false | 0 |
| strict | должна ли проверка учитывать соответствие типов данных trueValue или falseValu | false |

1. В UserController.php в actionCreate необходимо поменять redirect на /user и проверить добавление пользователей, для проверки запустить проект через OpenServer.
2. В UserController.php в actionCreate добавить Yii::$app->user->login($model) над строкой /user, в main.php заменить username на login и проверить действие авторизации после регистрации. Для проверки запустить проект через OpenServer

**Создание учетной записи Администратора, Авторизированного пользователя, Гостя**

1. Создать учетную запись Администратора в phpMyAdmin с логином «admin» и паролем «adminWSR» (в случае если согласно заданию демонстрационного экзамена потребуется учетная запись Администратора в БД).
2. Авторизованный пользователь появляется после заполненияя формы регистрации.
3. Учетная запись гость создана по умолчанию как пользователь без авторизации, поэтому создавать не нужно.

**Настройка виджета**

*Указания:* Студент настраивает виджет. Виджеты представляют собой конструкты, используемые для создания настраиваемых элементов пользовательского интерфейса. В комплект Yii2 входит большое количество виджетов.

Приведем пример настройки виджета «Навигация» как обязательного для выполнения согласно задания. Потребуется перейти в views/layouts/main.php. Настроить виджет Nav. Nav применяется для создания навигации на сайте.

 $items = [];

    if(Yii::$app->user->isGuest){

        $items[] =  ['label' => 'Регистрация', 'url' => ['/user/create']];

        $items[] =  ['label' => 'Авторизация', 'url' => ['/site/login']];

    } else {

        if(Yii::$app->user->identity->admin == 1){

            $items[] =  ['label' => 'Административная панель', 'url' => ['/admin']];

        } else{

            $items[] =  ['label' => 'Личный кабинет', 'url' => ['/user']];

        }

        $items [] = '<li class="nav-item">'

        . Html::beginForm(['/site/logout'])

        . Html::submitButton(

            'Logout (' . Yii::$app->user->identity->login . ')',

            ['class' => 'nav-link btn btn-link logout']

        )

        . Html::endForm()

        . '</li>';

    }

    echo Nav::widget([

        'options' => ['class' => 'navbar-nav'],

        'items' => $items,

**Создание раздела администратора (доступно только администратору)**

Создаем AdminController.php на основе UserController, переименовываем класс, удаляем все записи кроме actionIndex.

В папке views создаем папку admin и файл index.php в ней. Убираем GridView, меняем тайтл и добавляем ссылку на «Управление категориями».

**<?= Html::a('Управление категориями', ['/category']) ?>**

**Создание заявки**

*Указания*. Форма Заявки создается аналогично форме Регистрация, пункт «Настройка формы регистрации и запросы к валидатору». При добавлении Заявки она должна автоматически получить статус «Новая». Значение «Новая» формируется по умолчанию в ранее созданной таблице БД. При удалении заявки Авторизованному пользователю должно быть выведено сообщение с просьбой подтвердить желаемое действие. Для этого необходимо создать кнопку подтверждения удаления в панели управления.

Авторизованному пользователю должна быть доступна панель удаления на созданные им заявки. Пользователь может удалить только свою заявку, статус которой не был изменен Администратором на «решена» или «отклонена».

**Логотип**

*Указания*. Разработанный логотип должен быть сохранен на сервере в корневом каталоге в директории logo со следующим именем - logo.png в папке «проект студента»

1.Логотип должен быть реализован в соответствии с требованиями задания демонстрационного экзамена.

2. В логотипе должны быть использованы основные цвета сайта, необходимо наиболее часто встречающийся цвет и использовать его при создании логотипа.

3. Логотип представляет собой изображение, (paint, paint.net, photoshop, gimp, correl draw).



**Изменение картинки IMG**

*Указания*. Стиль анимации: «масштабирование». На картинке главной страницы необходимо разместить фотографию ПОСЛЕ (выгружает Администратор). После позиционирования указателя мыши на изображение ПОСЛЕ появляется начальное изображение ДО (загруженное Авторизованным пользователем. Размер изображения не должен превышать 10 МБ.

**Счетчик, плейер**

*Указания*. Информация о количестве решенных задачах обновляется автоматически, без

перезагрузки страницы не реже чем 1 раз в 5 секунд.

В случае изменения значения счетчика должно звучать оповещение (Notif.mp3).

Значение счетчика должно обновляться с анимацией.

**Новая функция в контроллере заявок:**

public function actionCount()

{

$count = Issue::find()

->where(['status' => 'Решена'])

->count();

return $count;

}

Возвращает количество заявок со статусом «Решена»

**Добавляем тег скрипт в index.php:**

<script src="js\updateSolvedIssues.js"></script>

Javascript подргужен.

**Код Javascript для обновления счетчика:**

document.addEventListener('DOMContentLoaded', function () {

$.ajax({

type: 'GET',

cache: false,

url: 'issue/count',

})

.done(function(data) {

document.getElementById("solved-number").innerHTML = "Решено уже: " + data + " проблем!";

})

.fail( function(data){

console.log('load went wrong!');

});

updateWithTimeout();

}, false);

function updateWithTimeout(){

$.ajax({

type: 'GET',

cache: false,

url: 'issue/count',

})

.done(function(data) {

var sd = $('#solved-number').text().replace(/[^0-9]/gi, '');

var current\_data = parseInt(sd, 10);

var new\_data = parseInt(data, 10);

if (new\_data > current\_data) {

const audio = new Audio('sounds/notif.wav');

audio.play();

}

document.getElementById("solved-number").innerHTML = "Решено уже: " + data + " проблем!";

console.log('updated!');

})

.fail( function(data){

console.log('load went wrong!');

});

setTimeout(updateWithTimeout, 5000);

}

С периодичностью в 5 секунд код асинхронно обращается к функции контроллера

И выводит количество заявок на странице.

Также при обновлении значения производится звук.

**Смена статуса заявки**

Заявке со статусом «Новая» можно сменить статус на «Решена» с обязательным

прикреплением (добавлением) фотографии – доказательства решения проблемы

(фотография «ПОСЛЕ»).

Заявке со статусом «Новая» можно сменить статус на «Отклонена» с

обязательным указанием причины отказа.

Смена статуса с «Решена» или «Отклонена» невозможна.

Zeal, загружаем в него документацию по Yii2 ищем метод beforeAction для web. Берем оттуда пример кода и вставляем в AdminController.php

Настройка OpenServer для работы с фреймворком

Настройка скриптов Yii2

Верстка

**Создание макета**

(Потребуется доступ к графическим редакторам paint (должен быть установлен)

Создание Страниц (Потребуется доступ к документации Zeal: php, html, css, bootstrap 5, laravel, jQuery, yii)

Backend (Выполняется на Yii2 или на Laravel (Требуется доступ к документации))

Тестирование и оптимизация web-приложения

Провести проверку оформления

Доступ к сетевой папке сервера для размещения папки domains-фио студента с выполненным проектом формат папки

Анимация настраивается индивидуально по желанию с применением Visual Code.

На главной странице изначально видна фотография «после». При наведении указателя мыши на фото «после», вместо него анимировано должна отображаться фотография «до».

Стиль анимации: «масштабирование». После вывода указателя мыши с изображения появляется начальное изображение «после» с тем же стилем анимации. На главной странице выводится не более 4 последних решенных проблем.

**Инструкция для участника**

Портал должен быть доступен по адресу: http://xxxxxx-m1.wsr.ru/, где xxxxxx –

логин пользователя, полученный участником на индивидуальное карточке

Административная панель должна быть доступна по адресу: http://xxxxxx

m1.wsr.ru/admin

**Внимание! Доступ ко всем страницам должен быть реализован с помощью**

**элементов пользовательского интерфейса.**

Медиа файлы располагаются на сервере в папке public. Вы можете изменять

предоставленные файлы и создавать новые для улучшения приложения.

Проверяются только работы, загруженные на сервер.

Ваш HTML/CSS должен быть валидным, а в консоли браузера и на сайте

отсутствовать ошибки/предупреждения/оповещения, создаваемые средствами

JavaScript и PHP/ аналог.

Оценка будет производиться в браузере Google Chrome или аналоге.