

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

Факультет информационных технологий
Кафедра «Инфокогнитивные технологии»

КУРСОВОЙ ПРОЕКТ

на тему: *«Приложение для анализа нарушений ПДД в регионах России»*
Направление подготовки 09.03.03 «Прикладная информатика»
Профиль «Корпоративные информационные системы»

Выполнила:

студентка группы 211-362

Меркель Ирина Сергеевна

21.01.2023

(подпись)

Москва 2023

Введение

На современном этапе развития автомобильный транспорт выступает важнейшим элементом транспортной системы страны. В связи с этим система обеспечения безопасности дорожного движения и эксплуатации транспортных средств приобретает все больше серьезных проблем, связанных с нарушением правил дорожного движения и эксплуатации транспортных средств.

Главной проблемой является тот факт, что нарушения ПДД происходят во всех регионах России ежедневно. Это несет угрозу для жизни и здоровья граждан, а также для инфраструктуры города.

Государство пытается бороться с нарушениями ПДД различными способами, например такими, как:

1. установка камер на дорогах;
2. введение новых видов штрафов;
3. усиление уже существующих штрафов и наказаний;
4. увеличение количества патрульных машин и др.

Но как показывает практика, существующих способов недостаточно для полного предотвращения нарушений.

Проект «Дорожная безопасность» поможет внести свой вклад в борьбу с нарушениями правил дорожного движения. Приложение использует открытые данные, связанные с дорожно-транспортными происшествиями, чтобы пользователь смог проанализировать их и получить рекомендации, как избежать те или иные виды ДТП и сохранить свое здоровье.

1 Цель и задачи работы

Целью работы является создание динамического и адаптированного под мобильные устройства веб-приложения, показывающего статистику нарушений дорожно-транспортных происшествий в регионах России, а также предоставляющего советы и рекомендации как для пешеходов, так и для водителей.

Этапы разработки приложения:

1. создать удаленный репозиторий на GitHub, подключить и описать файлы стилей, подключить библиотеку для динамических графиков, сверстать html-шаблоны для всех страниц приложения;
2. создать базу данных и подключиться к ней через приложение;
3. перенести открытые данные в независимые сущности в базе данных;
4. перенести html-верстку в файлы страниц формата .php;
5. написать логику для динамической загрузки данных, отобразить результат;
6. написать логику статистической обработки данных и отобразить результат на графиках;
7. написать советы и рекомендации для водителей и пешеходов;
8. выложить результат работы на хостинг.

Открытые данные содержат информацию про дорожно-транспортные происшествия¹ и безопасность дорожного движения². Исходные датасеты были очищены от лишних символов «пробел», а также была отредактирована первая строчка под названия атрибутов для загрузки в базу данных. В качестве базы данных была выбрана реляционная база данных MySQL, а для работы с динамикой и обработкой открытых данных языки программирования PHP и JavaScript.

2 Проектирование приложения

Структура веб-приложения состоит из следующих компонентов, которые написаны на языке программирования PHP:

1. `index.php` – главная страница приложения. Содержит краткое описание проекта, меню для выбора интересующей статистики или рекомендаций, а также ссылки на использованные открытые данные;
2. `regional_statistics.php` – страница, содержащая список для выбора регионов России, а также график со статистикой общего количества всех показателей датасета, график их процентного соотношения и график стандартного отклонения по каждому атрибуту;
3. `statistics_of_victims.php` – страница, содержащая список для выбора регионов России, а также график, отображающий статистику пострадавших в результате ДТП в выбранном ранее регионе, график с десятью регионами с наивысшим показателем ДТП и пострадавших и график стандартного отклонения по каждому атрибуту;
4. `tips_and_recommendations.php` – страница, содержащая меню для выбора типа пользователя, а также предоставляющая советы водителям и пешеходам;
5. `tips.php` – компонент, содержащий статический текст советов и рекомендаций в массиве;
6. `dbconnect.php` – компонент, содержащий код подключения к базе данных.

Реляционная модель базы данных представлена на рисунке 1.

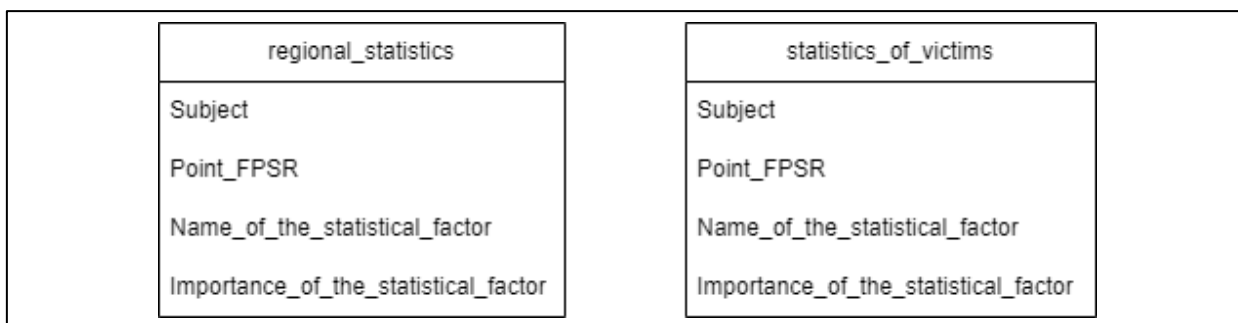


Рисунок 1 - Реляционная модель

Диаграмма компонентов приложения представлена на рисунке 2.

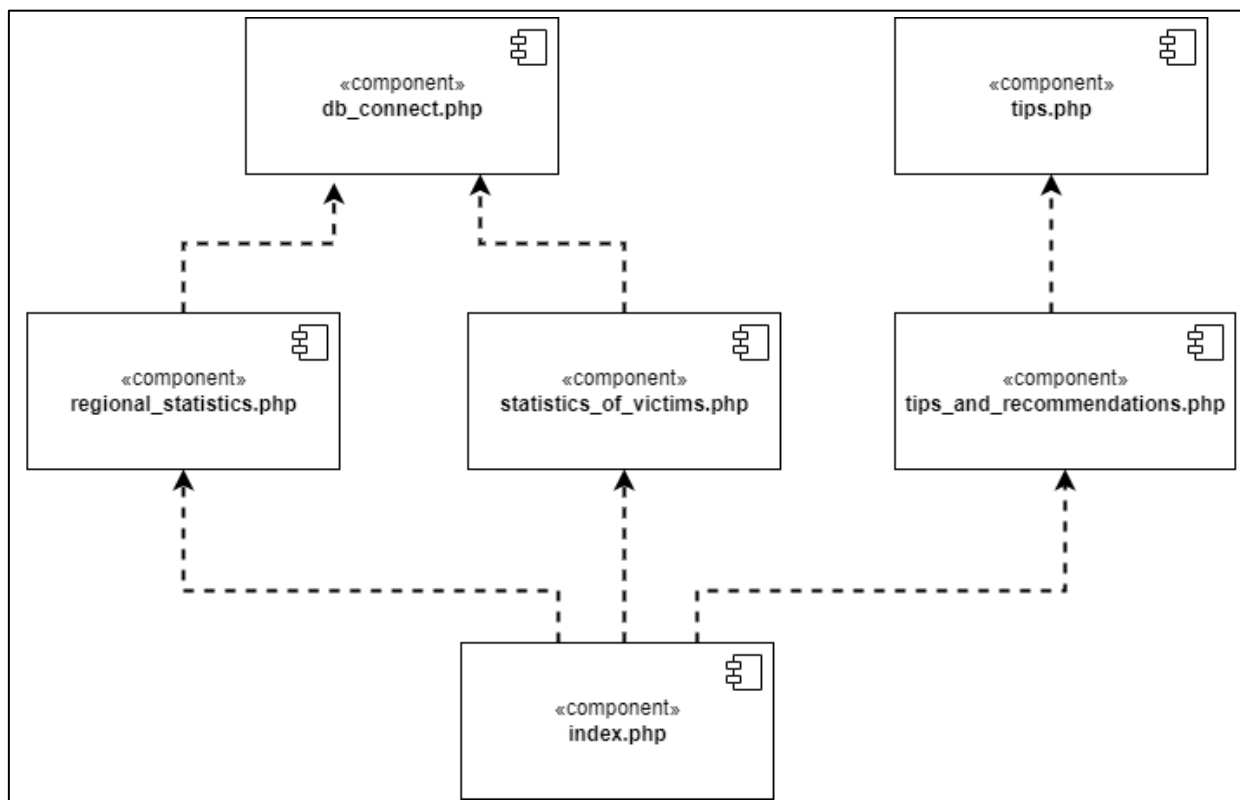


Рисунок 2 - Диаграмма компонентов

3 Реализация приложения

Разработка приложения началась с нахождения необходимых вспомогательных библиотек для построения графиков. Выбор пал на библиотеку AnyChart.js³. Далее были сверстаны макеты для будущих PHP-компонентов, а также описаны стили для каждого из них в отдельной директории. Все макеты адаптированы под мобильные устройства с помощью медиа запросов.

Готовые макеты были перенесены в файлы формата .php и связаны между собой ссылками. Затем была создана база данных MySQL на сервере московского политеха, в которую были выгружены открытые данные в независимые сущности (см. рисунок 1).

С помощью SELECT-запроса на страницах «статистика регионов» и «статистика пострадавших» был сформирован динамический список доступных для выбора регионов в виде массива, отсортированного по возрастанию значений при помощи встроенной функции sort, который после

использовался для вывода содержимого на страницу с помощью цикла. Элементы массива являются ссылками, передающие в атрибут «region» название региона.

Основным этапом стала реализация вычислительных фрагментов кода, которые использовали открытые данные для преобразований их в нужный формат, требуемый библиотекой anychart, исчислений процентного соотношения и стандартных отклонений показателей:

1. преобразование исходных открытых данных в необходимый для графика формат происходит с помощью SELECT-запроса нужного датасета. Результат на выходе является массивом, внутри которого находятся массивы из двух элементов, первым из которых является ключом, а второй – значением (см. приложение А п.1 и п.4).

2. преобразование исходных данных в процентное соотношение происходит с помощью формулы $\frac{s}{n} * 100$, где s – сумма всех показателей, n – их количество. Результат на выходе является массивом, внутри которого находятся массивы из двух элементов, первым из которых является ключом, а второй – значением (см. приложение А п.2).

3. вычисление стандартного отклонения для каждого из показателей необходимого датасета происходит по формуле $\sqrt{D} = \frac{\sum(x-\mu)^2}{n}$, где D – дисперсия, x – значение показателя, μ – среднее значение, n – количество показателей. Результат на выходе является массивом, внутри которого находятся массивы из двух элементов, первым из которых является ключом, а второй – значением (см. приложение А п.3 и п.5).

Страница с советами и рекомендациями была реализована с помощью отдельного php-компонента, который хранил статический текст в массиве. При выборе интересующего пункта в меню, которые являются ссылками, в запрос передается параметр со значением выбранного элемента меню и при помощи GET-массива подставляется как ключ в массив рекомендаций и отображает его содержимое на странице.

4 Основные сценарии использования приложения

Интерфейс главной страницы содержит меню с выбором интересующей статистики или рекомендаций для водителей и пешеходов, а также краткую информацию о приложении. Интерфейс изображен на рисунке 3.

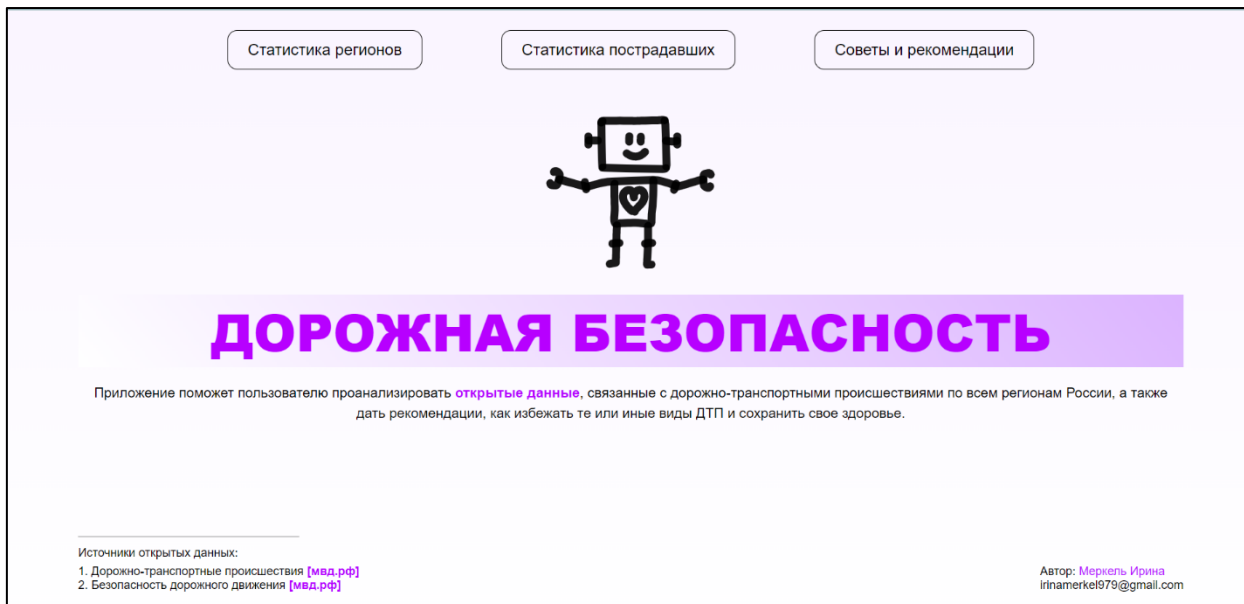


Рисунок 3 – Интерфейс главной страницы

При нажатии на кнопку «Статистика регионов» пользователь попадает на страницу, где нужно выбрать интересующий регион. На странице отображается статистика выбранного региона, связанная с дорожно-транспортными происшествиями, график процентного соотношения исходных данных, а также график со стандартным отклонением по каждому показателю. При наведении на столбец с показателем отображается название и количество данного атрибута. При нажатии на кнопки «Вернуться назад», «Статистика пострадавших» и «Советы и рекомендации» пользователь может переходить на соответствующие страницы. Интерфейс представлен на рисунке 4.

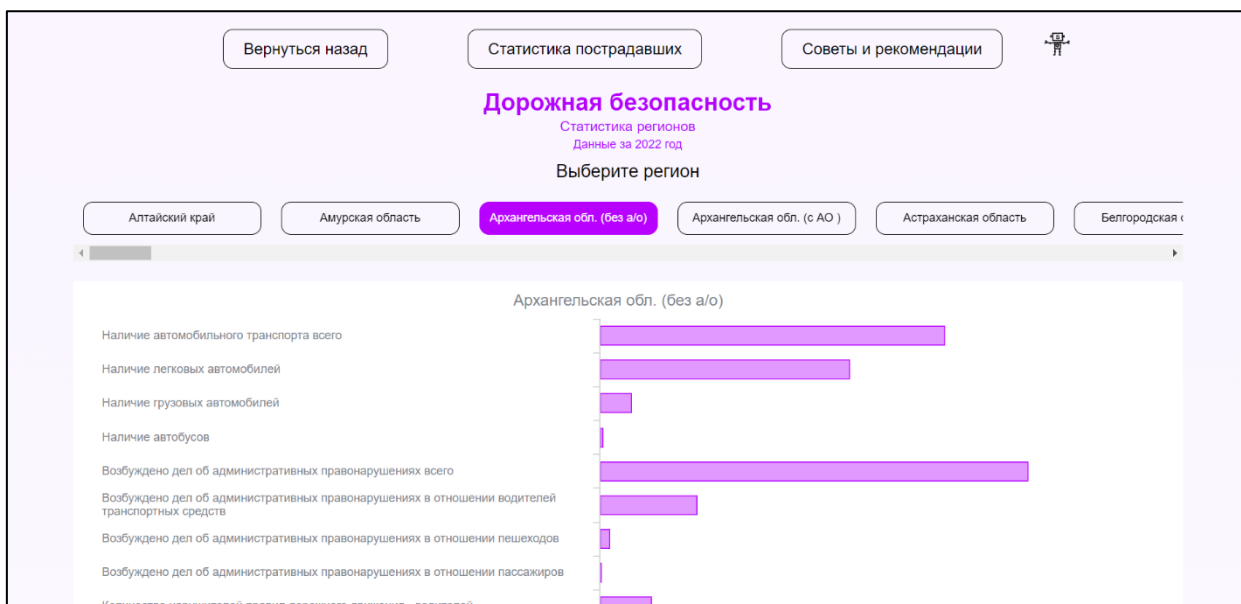


Рисунок 4 – Интерфейс для страницы со статистикой региона

При нажатии на кнопку «Статистика пострадавших» пользователь попадает на страницу, где нужно выбрать интересующий регион. На странице отображается статистика пострадавших во время ДТП, график с рейтингом из десяти регионов, лидирующих по количеству пострадавших в результате дорожно-транспортных происшествий, а также график со стандартным отклонением по каждому из показателей. При наведении на столбец с показателем отображается название и количество данного атрибута. При нажатии на кнопки «Вернуться назад», «Статистика регионов» и «Советы и рекомендации» пользователь может переходить на соответствующие страницы. Интерфейс представлен на рисунке 5.

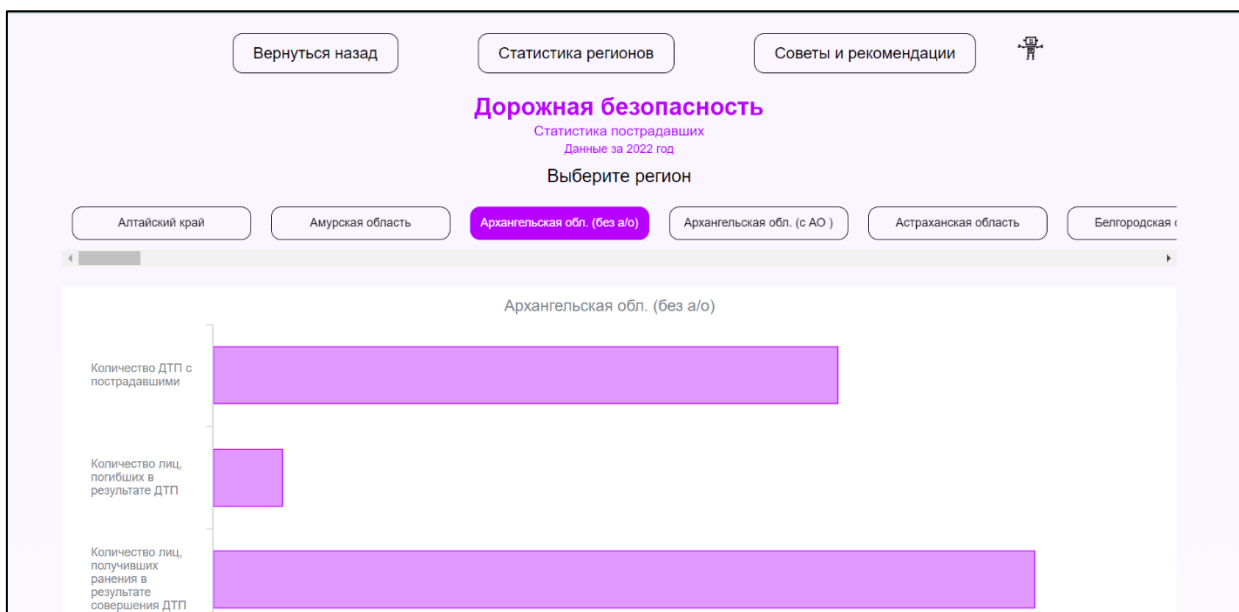


Рисунок 5 – Интерфейс для страницы со статистикой пострадавших

При нажатии на кнопку «Советы и рекомендации» пользователь попадает на страницу, где нужно выбрать категорию «Пешеход» или «Водитель». При выборе кнопки «Водитель» нужно указать тип погоды, который интересует пользователя. При выборе нужных параметров приложение показывает соответствующие рекомендации. При нажатии на кнопки «Возврат», «Статистика регионов» и «Статистика пострадавших» пользователь может переходить на соответствующие страницы. Интерфейс представлен на рисунке 6.

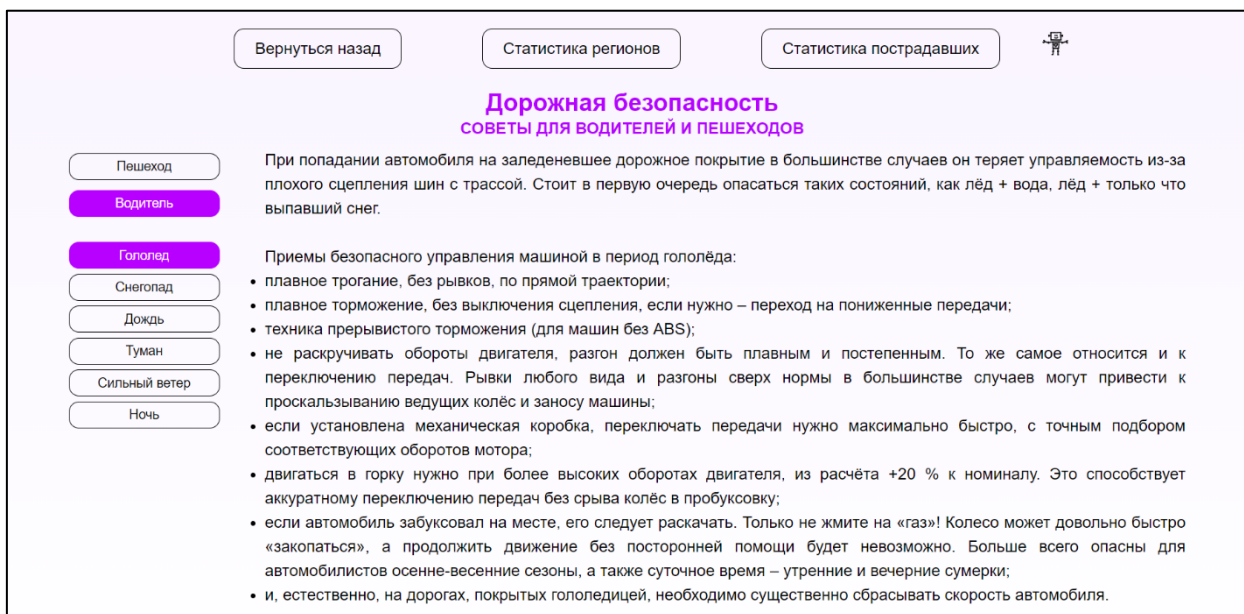


Рисунок 6 – Интерфейс для страницы с советами и рекомендациями

Заключение

В результате работы было создано динамическое и адаптивное веб-приложение, которое показывает статистику нарушений дорожно-транспортных происшествий во всех регионах России, дает возможность пользователю проанализировать открытые данные, а также посмотреть советы и рекомендации о поведении пешехода или водителя в различных погодных условиях. Приложение внесло свой вклад в развитие борьбы с нарушениями правил дорожного движения. Планируется расширять и дорабатывать данный проект в будущем.

Ссылка на веб-приложение, размещенное на хостинге: <http://traffic-assistant.std-2012.ist.mospolytech.ru/>

Ссылка на удаленный репозиторий приложения:
<https://github.com/Irinaa89/traffic-assistant>

Список литературы и интернет-ресурсов

1. Министерство Внутренних Дел Российской Федерации. Дорожно-транспортные происшествия. [Электронный ресурс]. URL: https://xn--b1aew.xn--p1ai/opendata/7727739372-MVD_GIAC_3.1 (дата обращения: 10.01.23).
2. Министерство Внутренних Дел Российской Федерации. Безопасность дорожного движения. [Электронный ресурс]. URL: <https://xn--b1aew.xn--p1ai/opendata/7727739372-MVDGIAC32> (дата обращения: 10.01.23).
3. AnyChart. Галерея Графиков И Диаграмм. [Электронный ресурс]. URL: <https://www.anychart.com/ru/products/anychart/gallery/> (дата обращения: 10.01.23).
4. Трасском. Безопасность дорожного движения в разных погодных условиях: советы по управлению автомобилем. [Электронный ресурс]. URL: <https://trasscom.ru/blog/bezopasnost-dorozhnogo-dvizheniya-v-raznyh-pogodnyh-usloviyah?ysclid=lcta40fzfd467086495#2> (дата обращения: 12.01.23).
5. Госавтоинспекция. Безопасность пешеходов. [Электронный ресурс]. URL: <https://xn--90adear.xn--p1ai/mens/pravo-peshehoda> (дата обращения: 12.01.23).

Приложение А

(справочное)

1. Листинг кода загрузки исходных открытых данных в динамический график на странице «Статистика регионов»:

```
if (isset($_GET["region"])) {  
    foreach ($regions as $value) {  
        if (str_replace(' ', '', $value) == $_GET["region"]) {  
            $query = mysqli_query($connect, "SELECT * FROM regional_statistics  
WHERE Subject='".$value.'\"");  
        }  
    }  
    while ($string = mysqli_fetch_assoc($query)) {  
        $arr = array();  
        array_push($arr, $string["Name_of_the_statistical_factor"]);  
        array_push($arr, (int)$string["Importance_of_the_statistical_factor"]);  
        array_push($data, $arr);  
    }  
}  
echo json_encode($data);
```

2. Листинг кода, преобразующий исходные открытые данные датасета для страницы «статистика регионов» в процентное соотношение:

```
$data = array();  
$region = "";  
if (isset($_GET["region"])) {  
    foreach ($regions as $value) {  
        if (str_replace(' ', '', $value) == $_GET["region"]) {  
            $query = mysqli_query($connect, "SELECT * FROM regional_statistics  
WHERE Subject='".$value.'\"");  
            $region = $value;  
        }  
    }  
    $percent_chart_keys = array();  
    while ($string = mysqli_fetch_assoc($query)) {  
        $arr = array();
```

```

        array_push($arr, $string["Name_of_the_statistical_factor"]);
        array_push($arr, (int)$string["Importance_of_the_statistical_factor"]);
        array_push($data, $arr);
        if (str_contains($string["Name_of_the_statistical_factor"], "всего")) {
            $percent_chart_keys[$string["Name_of_the_statistical_factor"]] =
(int)$string["Importance_of_the_statistical_factor"];
        }
    }
}
$percent_chart_data = array();
if (isset($_GET["region"])) {
    $query = mysqli_query($connect, "SELECT * FROM regional_statistics WHERE
Subject='".$_region."'");
    while($string = mysqli_fetch_assoc($query)) {
        if (str_contains($string["Name_of_the_statistical_factor"], "Наличие") &&
!str_contains($string["Name_of_the_statistical_factor"], "всего")) {
            $arr = array();
            array_push($arr, $string["Name_of_the_statistical_factor"]);
            array_push($arr, round($string["Importance_of_the_statistical_factor"] /
$percent_chart_keys["Наличие автомобильного транспорта всего"] * 100, 2));
            array_push($percent_chart_data, $arr);
        }
        if (str_contains($string["Name_of_the_statistical_factor"], "Возбуждено") &&
!str_contains($string["Name_of_the_statistical_factor"], "всего")) {
            $arr = array();
            array_push($arr, $string["Name_of_the_statistical_factor"]);
            array_push($arr, round($string["Importance_of_the_statistical_factor"] /
$percent_chart_keys["Возбуждено дел об административных правонарушениях всего"]*
100, 2));
            array_push($percent_chart_data, $arr);
        }
        if (str_contains($string["Name_of_the_statistical_factor"], "Количество") &&
!str_contains($string["Name_of_the_statistical_factor"], "всего")) {
            $arr = array();
            array_push($arr, $string["Name_of_the_statistical_factor"]);

```

```

        array_push($arr, round($string["Importance_of_the_statistical_factor"] /
$percent_chart_keys["Количество нарушителей правил дорожного движения всего"] * 100,
2));

        array_push($percent_chart_data, $arr);
    }
}
}

echo json_encode($percent_chart_data);

```

3. Листинг кода, высчитывающий среднее отклонение для каждого показателя в датасете на странице «Статистика регионов»:

```

$data = array();

$query = mysqli_query($connect, "SELECT * FROM regional_statistics");
// Подсчет суммы по регионам
$regions_sum = array();
while ($string = mysqli_fetch_assoc($query)) {
    if ($string["Subject"] != "Всего по России") {
        if (isset($regions_sum[$string["Name_of_the_statistical_factor"]])) {
            $regions_sum[$string["Name_of_the_statistical_factor"]] +=
$string["Importance_of_the_statistical_factor"];
        } else {
            $regions_sum[$string["Name_of_the_statistical_factor"]] =
$string["Importance_of_the_statistical_factor"];
        }
    }
}

// Подсчет среднего значения
$regions_avg = array();
foreach($regions_sum as $key => $value) {
    $regions_avg[$key] = round($regions_sum[$key] / (count($regions) - 1), 2);
}

mysqli_data_seek($query, 0);
$dispersion = array();
$numerator = array();
while($string = mysqli_fetch_assoc($query)) {
    if ($string["Subject"] != "Всего по России") {

```

```

foreach($regions_avg as $key => $value) {
    if($key == $string["Name_of_the_statistical_factor"]) {
        if (isset($numerator[$key])) {
            $numerator[$key] += pow($string["Importance_of_the_statistical_factor"] -
$value, 2);

            $answer = round(sqrt($numerator[$key] / (count($regions) - 1)), 2);
            $dispersion[$key] = $answer;
        } else {
            $numerator[$key] = pow($string["Importance_of_the_statistical_factor"] -
$value, 2);
        }
    }
}
}
}
}
}
foreach($dispersion as $key => $value) {
    $arr = array();
    array_push($arr, $key);
    array_push($arr, $value);
    array_push($data, $arr);
}
echo json_encode($data);

```

4. Листинг кода выгрузки исходных открытых данных в динамический график на странице «Статистика пострадавших»:

```

$data = array();
if (isset($_GET["region"])) {
    foreach ($regions as $value) {
        if (str_replace(' ', '', $value) == $_GET["region"]) {
            $query = mysqli_query($connect, "SELECT * FROM statistics_of_victims
WHERE Subject='".$value."'");
        }
    }
}
while ($string = mysqli_fetch_assoc($query)) {
    $arr = array();
    array_push($arr, $string["Name_of_the_statistical_factor"]);
}

```

```

        array_push($arr, (int)$string ["Importance_of_the_statistical_factor"]);
        array_push($data, $arr);
    }
}
echo json_encode($data);

```

5. Листинг кода, высчитывающий стандартное отклонение показателей в датасете для страницы «Статистика пострадавших»:

```

$data = array();
$query = mysqli_query($connect, "SELECT * FROM statistics_of_victims");
// Подсчет суммы по регионам
$regions_sum = array();
while ($string = mysqli_fetch_assoc($query)) {
    if ($string["Subject"] != "Всего по России") {
        if (isset($regions_sum[$string["Name_of_the_statistical_factor"]])) {
            $regions_sum[$string["Name_of_the_statistical_factor"]] += (int)
$string["Importance_of_the_statistical_factor"];
        } else {
            $regions_sum[$string["Name_of_the_statistical_factor"]] = (int)
$string["Importance_of_the_statistical_factor"];
        }
    }
}
// Подсчет среднего значения
$regions_avg = array();
foreach($regions_sum as $key => $value) {
    $regions_avg[$key] = round($regions_sum[$key] / (count($regions) - 1), 2);
}
mysqli_data_seek($query, 0);
$dispersion = array();
$numerator = array();
while($string = mysqli_fetch_assoc($query)) {
    if ($string["Subject"] != "Всего по России") {
        foreach($regions_avg as $key => $value) {
            if($key == $string["Name_of_the_statistical_factor"]) {
                if (isset($numerator[$key])) {

```



```

        $numerator[$key] += pow((int)
$string["Importance_of_the_statistical_factor"] - $value, 2);
        $answer = round(sqrt($numerator[$key] / (count($regions) - 1)), 2);
        $dispersion[$key] = $answer;
    } else {
        $numerator[$key] = pow((int) $string["Importance_of_the_statistical_factor"]
- $value, 2);
    }
}
}
}
}
}
foreach($dispersion as $key => $value) {
    $arr = array();
    array_push($arr, $key);
    array_push($arr, $value);
    array_push($data, $arr);
}
echo json_encode($data);

```