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: *«Веб-приложение для анализа статистики преступности в регионах России»*

Направление подготовки 09.03.03 «Прикладная информатика»

Профиль «Корпоративные информационные системы»

**Выполнил:**

студент группы 211-362

Шарков Иван Александрович

|  |  |
| --- | --- |
| 19.01.2023 |  |
|  | (подпись) |

Москва 2023

**Введение**

Преступность зародилась вместе с обществом и существует по сегодняшний день. Эволюция преступности — это одна из составляющих истории развития общества, обхватывающая все существующие сферы жизни людей. Преступность существовала во все времена, изменяясь не только от эпохи к эпохе, от страны к стране, но и, пусть и редко, от региона к региону.

Самая главная проблема преступности — угроза жизни и здоровью граждан Российской Федерации.

В наше время существует немало путей решения проблемы с преступностью, например:

* неотвратимость и суровость наказания;
* достижение невыгодности совершения преступления;
* техническая оснащенность противостоящих органов;
* привлечение международного сообщества;
* заинтересованность общества.

Но всех этих путей решения недостаточно, ведь преступления совершаются ежедневно, в разных точках мира и в немалых количествах, что подталкивает общество искать новые пути борьбы с данной проблемой.

Веб-приложение «Наша безопасность» — это один из новых видов путей решения проблем с преступностью. Приложение помогает пользователю анализировать открытые данные МВД РФ по совершенным, зарегистрированным и раскрытым преступлениям, отображает статистические данные на графиках для интересующего региона и предоставляет рекомендации гражданам Российской Федерации о том, как избежать и снизить риск сталкивания с нарушителями закона.

**1 Цель и задачи работы**

Целью работы является создание адаптивного, динамического веб-приложения, которое помогает пользователю анализировать открытые данные по совершенным преступлениям, отображает обработанные данные на графиках и предоставляет полезные советы и рекомендации по борьбе с преступностью гражданам Российской Федерации.

Основные этапы разработки приложения:

1. сверстать HTML-структуру основных страниц, описать стили, подключить сторонние вспомогательные библиотеки;
2. создать локальную базу данных, заполнить ее датасетами, настроить подключение с базой данных, протестировать взаимодействие с приложением;
3. выгрузить данные из локальной базы данных в приложение, создать на их основе динамические кнопки для выбора региона и опций графика;
4. написать функции-обработчики, которые обработают открытые данные, преобразуют их в нужный формат и перенесут результат на графики;
5. определить рейтинг самых частых нарушений УК РФ и на его основе предоставить советы и рекомендации для каждого пункта;
6. Разместить приложение на хостинге Московского Политеха.

Исходные открытые данные содержат информацию о зарегистрированных, раскрытых и нераскрытых преступлениях1, информацию о преступлениях, по которым имеются потерпевшие2, информацию о преступности3 и имеют одинаковые атрибуты. В качестве средств разработки используются языки программирования PHP и JavaScript для динамических вычислений и функционала приложения, язык гипертекстовой разметки HTML и препроцессор SCSS4 для языка стилей CSS для интерфейса приложения.

**2 Проектирование приложения**

Структура разработанного проекта состоит из компонентов, написанных на языке PHP. Компоненты поделены на следующие смысловые блоки:

***Блок компонентов-страниц:***

1. index.php – титульная страница приложения, которая содержит информацию о проекте, ссылки на использованные открытые данные, а также кнопку для перехода к работе на главной странице приложения;
2. main.php – главная страница приложения, которая содержит графики, динамические кнопки для выбора региона и соответствующей для данного региона опции. Содержит JavaScript-код для динамического создания и настройки графиков на странице. Для реализации графиков используется библиотека chart.js. Данные на графики динамически подставляются при помощи PHP-кода;
3. recommends.php – страница с пятью наиболее часто нарушаемыми статьями УК РФ, содержащими ссылки на рекомендации по каждому пункту, а также ссылки на использованную литературу. Список статей формируется динамически в зависимости от ранее выбранного региона на главной странице. Каждый элемент списка является ссылкой на страницу с рекомендациями по данной статье;
4. article\_recommends.php – страница с рекомендацией по конкретной статье уголовного кодекса с ссылками на использованную литературу. Данные динамически подставляются в зависимости от выбранной ранее статьи.

***Блок компонентов-утилит:***

1. stats.php – компонент, содержащий основные вычислительные функции для обработки открытых данных. Содержит функции для подсчета количества данных и их процентного отношения для общей статистики, функции для подсчет количества данных и их процентного отношения для конкретного датасета, функцию для вычисления стандартного отклонения для каждого из показателей конкретного датасета;
2. config.php – компонент, служащий для подключения приложения к базе данных;
3. querys.php – компонент, содержащий основные запросы к базе данных;
4. regions.php – компонент, обрабатывающий открытые данные для создания массива-списка регионов Российской Федерации;
5. articles.php – компонент, содержащий статические переменные с рекомендациями для каждой статьи уголовного кодекса.

Диаграмма компонентов, отображающая специфику взаимодействия между компонентами веб-приложения, написанная на языке диаграмм UML, представлена на рисунке 1.

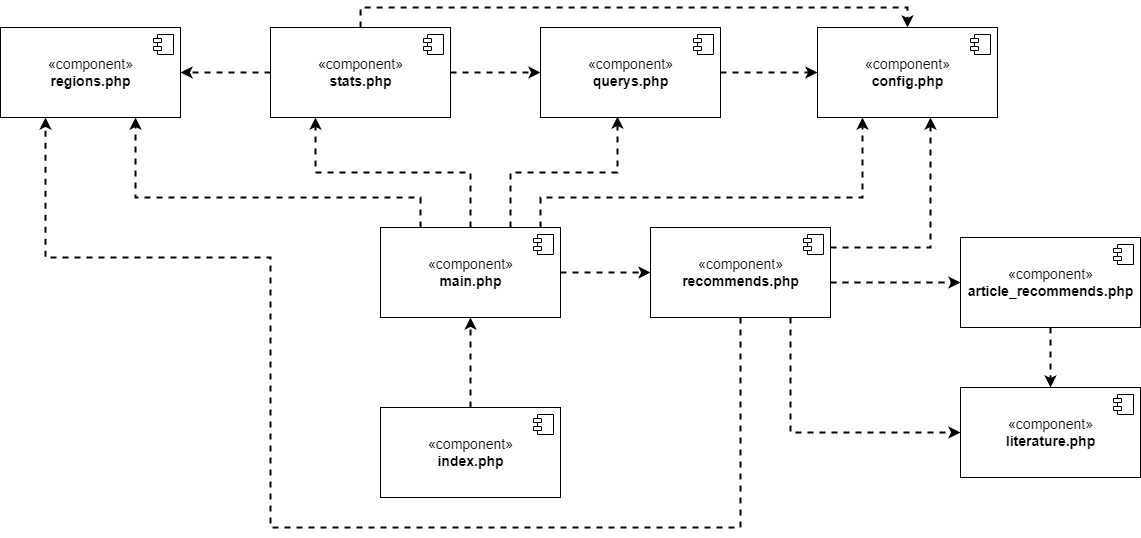


Рисунок 1 – Диаграмма компонентов веб-приложения

Для хранения открытых данных была выбрана база данных MySQL. Перед импортом данных были исправлены опечатки с лишними пробелами в csv файлах, а также первая строчка данных была сразу подкорректирована под желаемые названия атрибутов. В результате получились три независимые отношения. Реляционная модель базы данных представлена на рисунке 2.

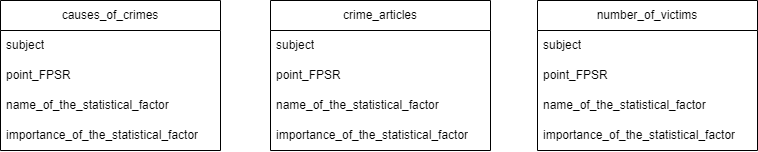


Рисунок 2 – Реляционная модель базы данных

Для разработки интерфейса использовался препроцессор SCSS для более быстрого и удобного написания стилей, а также построения понятной и красивой архитектуры. Также при разработке использовалась популярная библиотека готовых стилей bootstrap55. Основное предназначение библиотеки пришлось на использование адаптивной сетки для приложения. Приложение имеет динамический задний фон, написанный на языке JavaScript с помощью встроенного класса Canvas6. Веб-приложение полностью адаптировано под мобильные устройства с помощью медиа запросов и сетки bootstrap5.

**3 Реализация приложения**

Первым этапом разработки приложения стала необходимость создания удаленного репозитория на платформе GitHub, подключения созданного проекта к базе данных MySQL, настройки стартового шаблона для начала проектирования. Верстка выполнена при помощи технологии «flex» с использованием библиотеки bootstrap5, которая предоставила адаптивную сетку, а препроцессор SCSS помог выстроить качественную и понятную для других разработчиков архитектуру стилей всего веб-приложения благодаря поддержке модульного подключения.

После простых обработок открытых данных, таких как удаление лишних пробелов, датасеты были выгружены в базу данных в независимые отношения (см. рисунок 2).

Далее были созданы все необходимые страницы в формате .php из сверстанных html-файлов с подключенными библиотеками bootstrap5 и chart.js. Прописаны медиа запросы для адаптации верстки под мобильные устройства.

Готовые html-шаблоны были перенесены в php-компоненты для дальнейшей работы. После успешной настройки подключения приложения к базе данных в компоненте config.php, с помощью sql-запроса к одному из датасетов был сформирован массив со всеми регионами России. Массив регионов был выгружен на страницу в качестве кнопок для выбора, отсортированных в алфавитном порядке.

Следующий этап был написанием основных вычислительных функций для обработки открытых данных. В результате было написано 5 функций, которые высчитывали количественные значения атрибутов региона, их процентное соотношение и стандартное отклонение по каждому из них. Для статистической обработки открытых данных используются функции, реализованные в компоненте stats.php.

Функция ***count\_general\_statistics*** высчитывает общее количество атрибутов со всех датасетов с помощью sql-запроса и возвращает массив данных формата ключ => значение (см. приложение А п. 1).

Функция ***count\_general\_statistics\_percent*** высчитает процентное соотношение атрибутов со всех датасетов для общей статистики по формуле \* 100, где s – сумма, n – количество и возвращает массив данных формата ключ => значение. (см. приложение А п. 2).

Функция ***count\_quantitative\_values*** выгружает исходные открытые данные конкретного региона для конкретной опции с помощью sql-запроса, приводя их к виду, подходящего для загрузки содержимого на график в формате ключ => значение (см. приложение А п. 3).

Функция ***count\_percent\_values*** высчитывает процентное соотношение показателей для конкретного региона по формуле \* 100, где s – сумма, n – количество и возвращает массив данных формата ключ => значение (см. приложение А п.4).

Функция ***count\_dispersion*** высчитывает стандартное отклонение по каждому атрибуту для конкретного датасета по формуле стандартного отклонения = , где D – дисперсия, x – значение показателя, – среднее значение всех данных, n – количество данных и возвращает массив формата ключ => значение (см. приложение А п.5).

Заключительный этап стал созданием рейтинга наиболее часто нарушаемых уголовных кодексов в конкретном регионе. Результат sql-запроса для отношения с датасетом, содержащий статистику нарушений уголовных кодексов, был отсортирован по количеству значений каждого атрибута и обрезан до 5-го элемента включительно. Элементы рейтинга ссылаются на страницу с советами и рекомендациями по выбранному уголовному кодексу.

Разработанный и протестированный проект был выложен на хостинг Московского Политеха для общего доступа.

**4 Основные сценарии использования приложения**

Интерфейс титульной страницы содержит краткое описание веб-приложения, ссылки на использованные открытые данные, а также кнопку «Выбрать регион», которая перемещает пользователя на главную страницу приложения. Интерфейс титульной страницы представлен на рисунке 3.

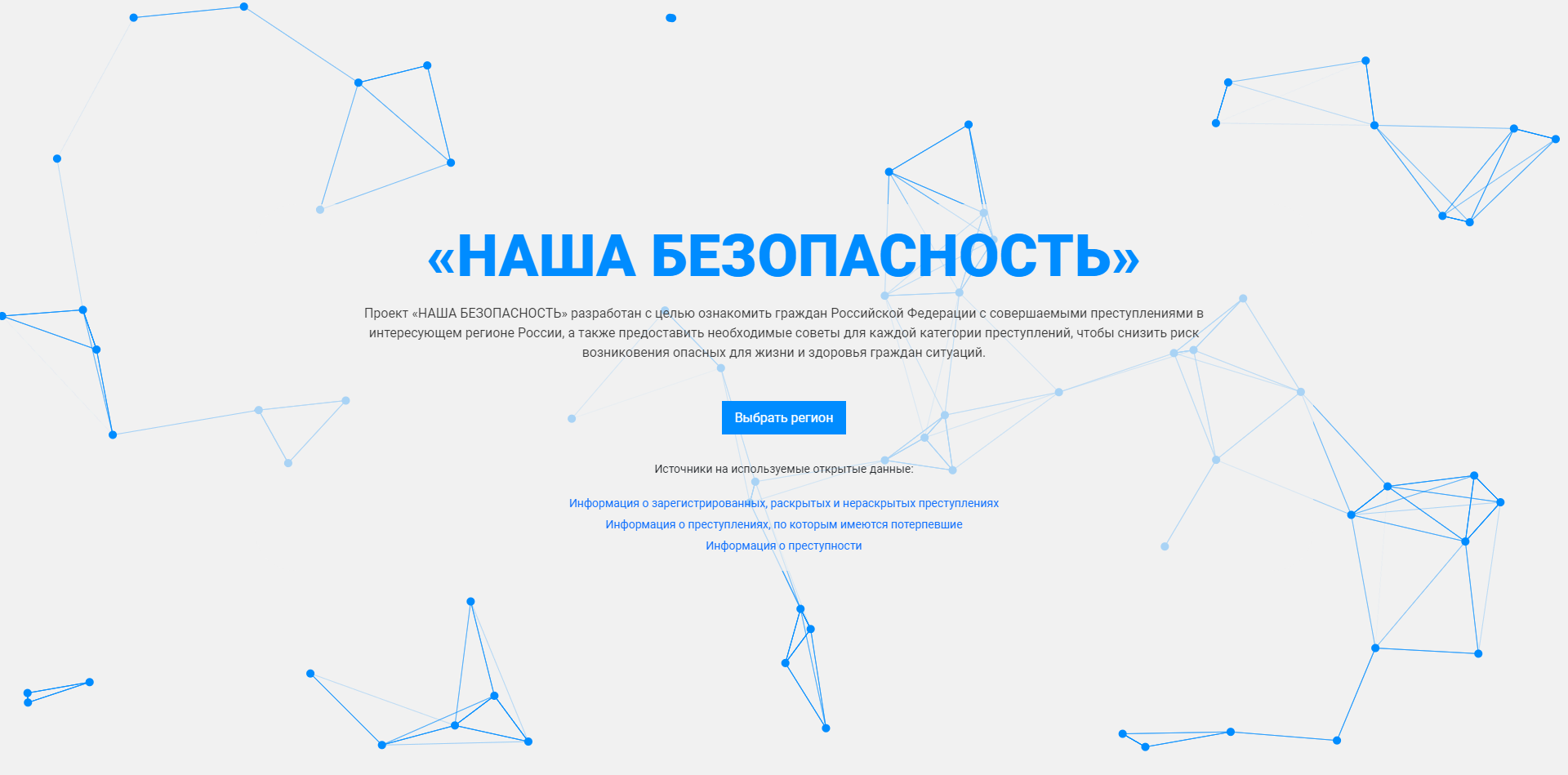


Рисунок 3 – Интерфейс титульной станицы веб-приложения

После нажатия на кнопку «Выбрать регион», пользователь перемещается на главную страницу. Главная страница содержит три основных блока: блок с кнопками для выбора региона, блок с динамическими графиками, блок с доступными для выбора опциями. При отсутствии выбранных параметров, графики не отображаются.

Для общей статистики представлен график общей статистики на основании всех датасетов, а также их процентное соотношение. Для каждого конкретного датасета реализуется три графика: количественные показатели, процентное соотношение данных показателей и стандартное отклонение по каждому из атрибутов. При наведении на элемент графика отображается текст с названием атрибута и его количества. Интерфейс главной страницы с невыбранными параметрами представлен на рисунке 4.

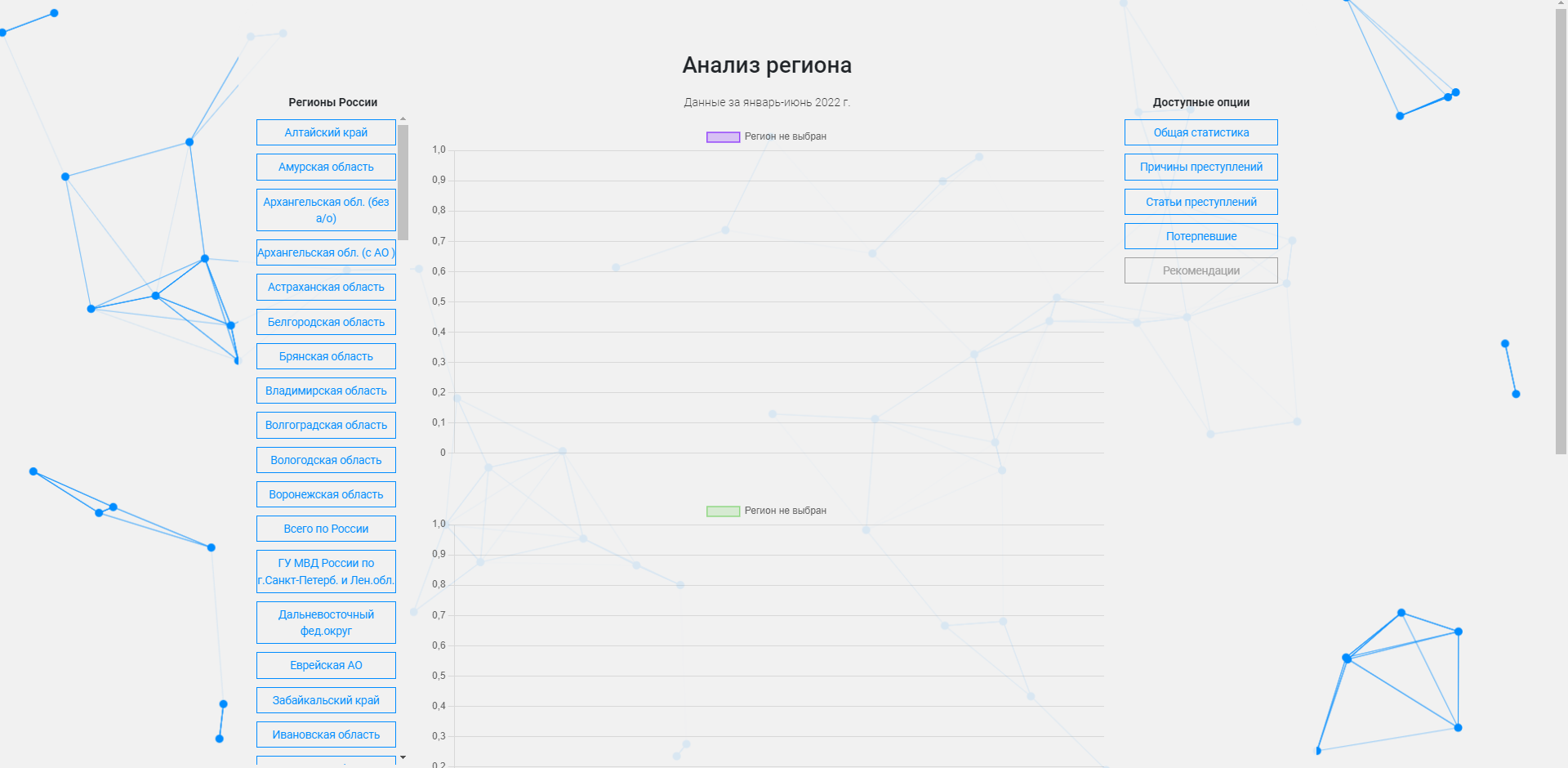


Рисунок 4 – Интерфейс главной страницы веб-приложения с невыбранными параметрами

При выборе необходимых параметров пользователем, динамически строятся статистические графики, а также становится доступным для нажатия кнопка «Рекомендации» для выбранного региона. Интерфейс с выбранным регионом и соответствующей для него опции представлен на рисунке 5 и рисунке 6.

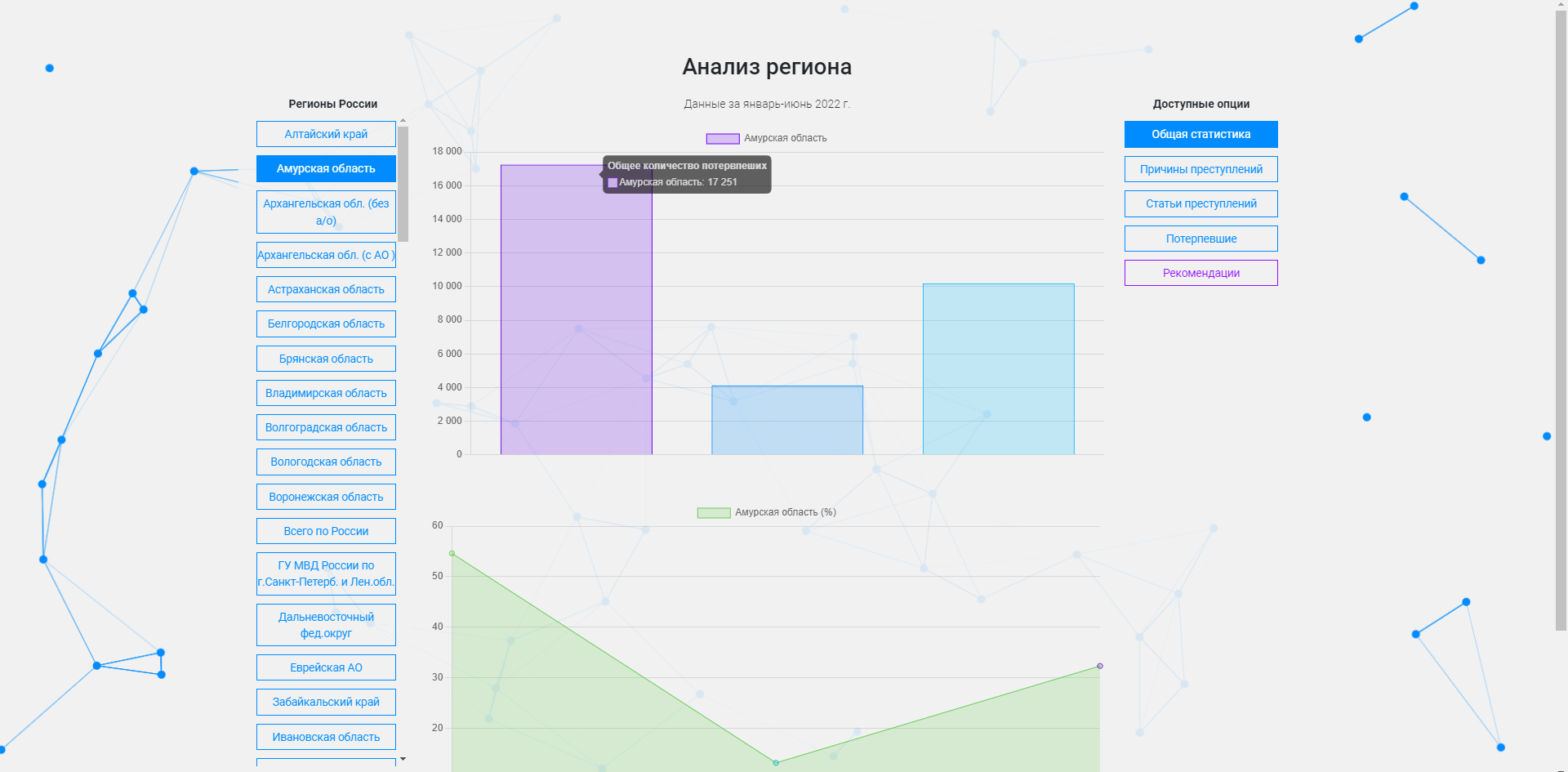


Рисунок 5 – Интерфейс главной страницы веб-приложения с выбранными параметрами



Рисунок 6 График стандартного отклонения показателей

При нажатии на кнопку «Рекомендации» пользователь перейдет на страницу со списком наиболее часто нарушаемых статей по ранее выбранному региону. Каждый элемент списка является ссылкой на страницу с рекомендациями по данному элементу. Также снизу приведены источники на литературу, используемую для написания советов, и рекомендации. Интерфейс страницы с наиболее часто нарушаемыми статьями представлен на рисунке 7.

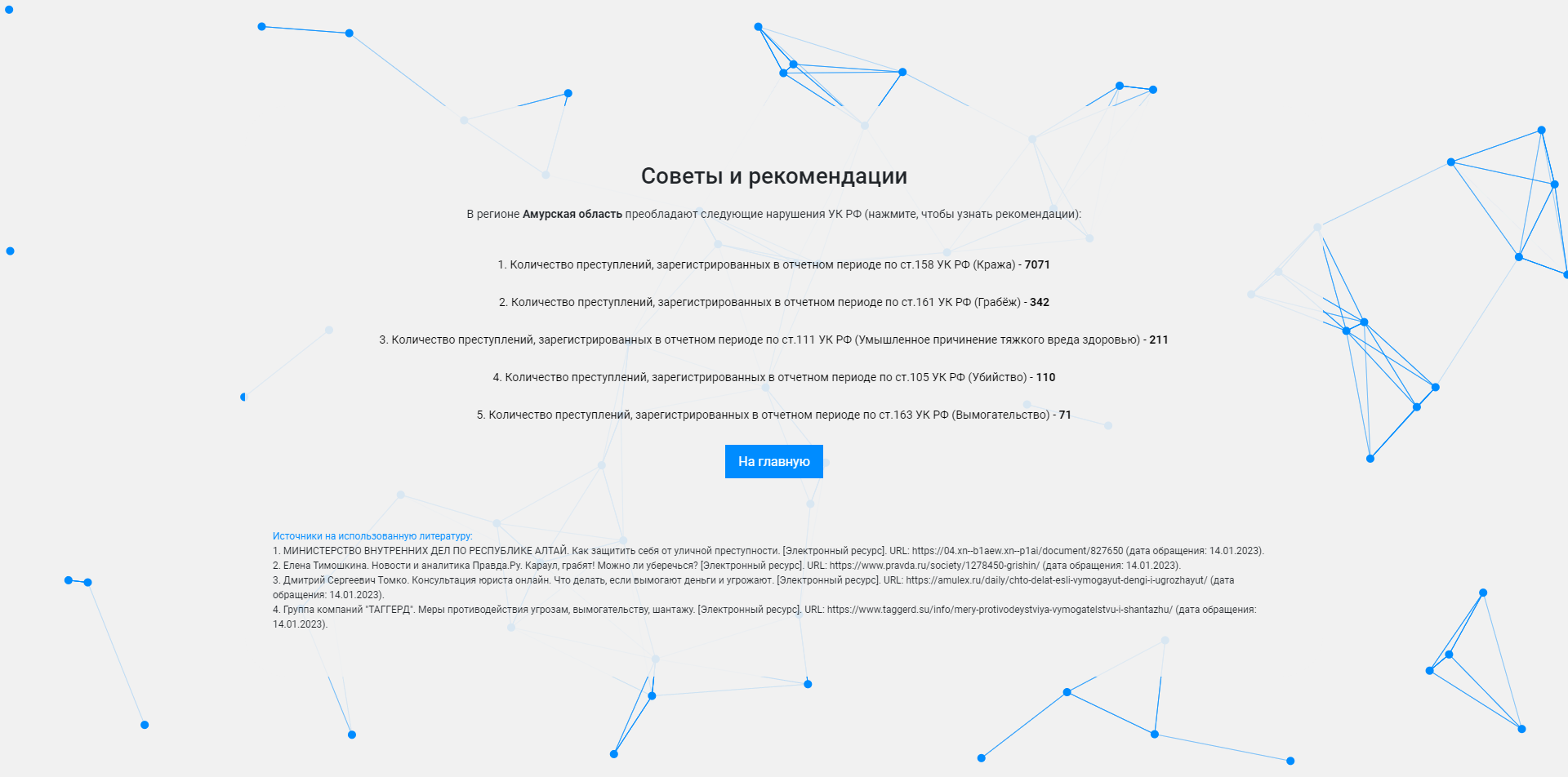


Рисунок 7 – Интерфейс страницы с наиболее часто нарушаемыми статьями в регионе

После выбора интересующей статьи, пользователь попадает на страницу с рекомендациями и советами по данному уголовному кодексу. Интерфейс страницы с советами и рекомендациями представлен на рисунке 8.

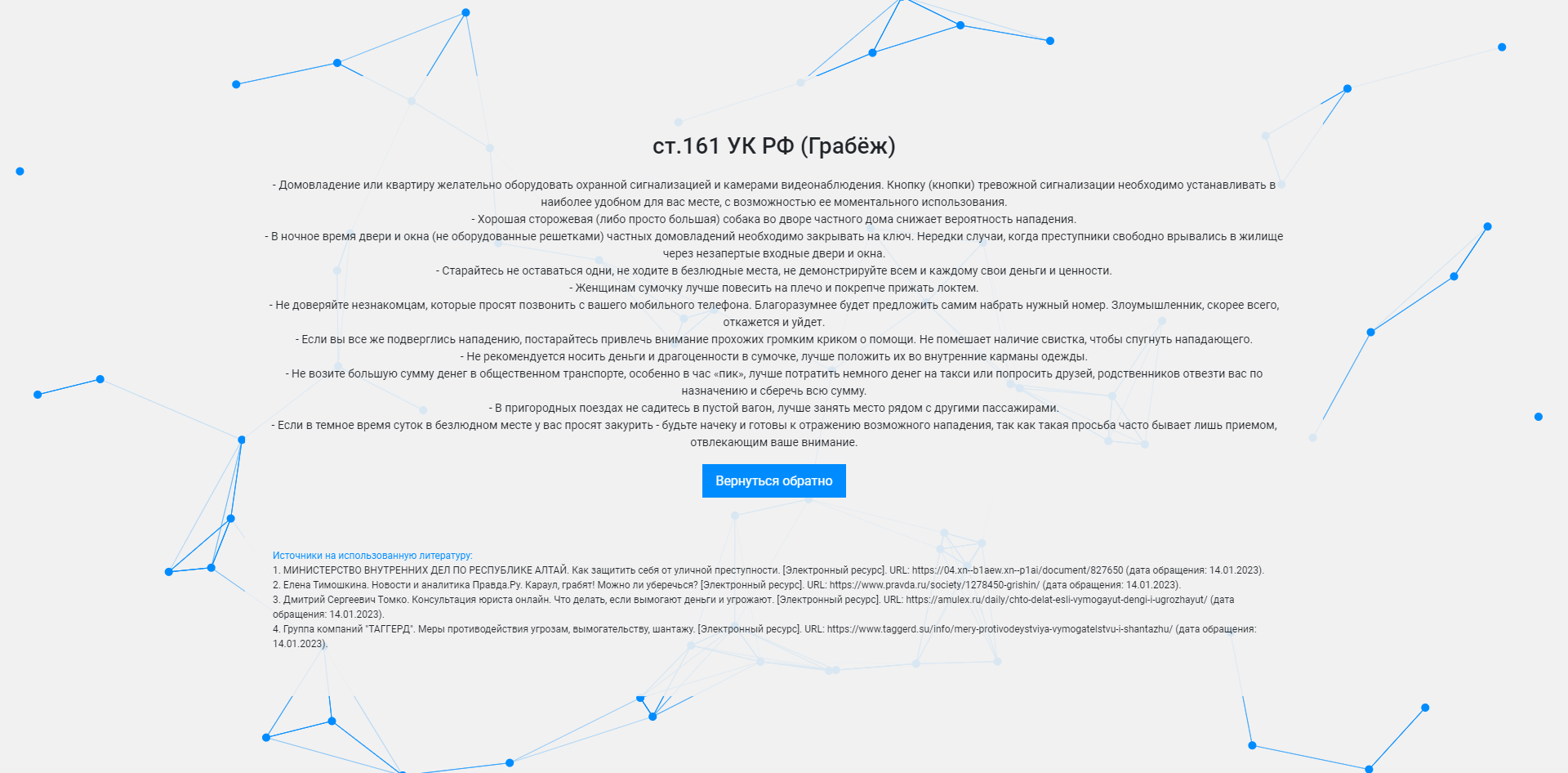


Рисунок 8 – Интерфейс страницы с советами и рекомендациями по выбранной статье

**Заключение**

Результатом работы является динамическое и адаптивное веб-приложение, способное обрабатывать исходные открытые данные по статистике преступлений в регионах России, а также демонстрирующее результаты обработки на динамических графиках и рекомендации для снижения риска сталкивания с противоправными ситуациями. Проект внес свой вклад в борьбу с преступностью в стране. В будущем приложение будет дорабатываться, расширяться и улучшаться.

Ссылка на веб-приложение: http://our-safety.std-2021.ist.mospolytech.ru/.

Ссылка на удаленный репозиторий проекта: https://github.com/scharkoff/our-safety.

**Список литературы и интернет-ресурсов**

1. Министерство Внутренних Дел Российской Федерации. Информация о зарегистрированных, раскрытых и нераскрытых преступлениях. [Электронный ресурс]. URL: https://xn--b1aew.xn--p1ai/opendata/7727739372-MVDGIAC38 (дата обращения 10.01.2023).
2. Министерство Внутренних Дел Российской Федерации. Информация о преступлениях, по которым имеются потерпевшие. [Электронный ресурс]. URL: https://xn--b1aew.xn--p1ai/opendata/7727739372-MVDGIAC310 (дата обращения 10.01.2023).
3. Министерство Внутренних Дел Российской Федерации. Информация о преступности. [Электронный ресурс]. URL: https://xn--b1aew.xn--p1ai/opendata/7727739372-MVDGIAC33 (дата обращения 10.01.2023).
4. Министерство Внутренних Дел Российской Федерации. Информация о зарегистрированных, раскрытых и нераскрытых преступлениях. [Электронный ресурс]. URL: https://xn--b1aew.xn--p1ai/opendata/7727739372-MVDGIAC38 (дата обращения 10.01.2023).
5. Русская документация и примеры по самой популярной в мире библиотеке HTML, CSS и JS - Bootstrap 5. [Электронный ресурс]. URL: https://bootstrap5.ru/docs/getting-started/introduction (дата обращения 10.01.2023).
6. Beautiful HTML5 Charts & Graphs. [Электронный ресурс]. URL: https://canvasjs.com/ (дата обращения 10.01.2023).

**Приложение А**

*(справочное)*

Программный код основных вычислительных функций

1. Программный код функции **count\_general\_statistics** на языке программирования PHP:

function count\_general\_statistics($region) {

$result = array();

global $connect;

global $crime\_articles;

global $causes\_of\_crimes;

global $number\_of\_victims;

// -- Return arrow to start of query string result

mysqli\_data\_seek($crime\_articles, 0);

mysqli\_data\_seek($causes\_of\_crimes, 0);

mysqli\_data\_seek($number\_of\_victims, 0);

// -- Sum total amount for every name of the statistical factor of the current region:

// -- Add to result array new values [$key => name of the statistical factor, $value => sum of this statistical factor]

while ($row = mysqli\_fetch\_assoc($number\_of\_victims)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

if (isset($result["Общее количество потервпеших"])) {

$result["Общее количество потервпеших"] += $row["importance\_of\_the\_statistical\_factor"];

} else {

$result["Общее количество потервпеших"] = $row["importance\_of\_the\_statistical\_factor"];

}

}

}

// -- Add to result array new values [$key => name of the statistical factor, $value => sum of this statistical factor]

while ($row = mysqli\_fetch\_assoc($causes\_of\_crimes)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

if (isset($result["Алкогольные, токсические, наркотические опьянения или лица, ранее совершавшие преступления"])) {

$result["Алкогольные, токсические, наркотические опьянения или лица, ранее совершавшие преступления"] += $row["importance\_of\_the\_statistical\_factor"];

} else {

$result["Алкогольные, токсические, наркотические опьянения или лица, ранее совершавшие преступления"] = $row["importance\_of\_the\_statistical\_factor"];

}

}

}

// -- Add to result array new values [$key => name of the statistical factor, $value => sum of this statistical factor]

while ($row = mysqli\_fetch\_assoc($crime\_articles)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

if (isset($result["Общее число нарушений УК РФ"])) {

$result["Общее число нарушений УК РФ"] += $row["importance\_of\_the\_statistical\_factor"];

} else {

$result["Общее число нарушений УК РФ"] = $row["importance\_of\_the\_statistical\_factor"];

}

}

}

return $result;

}

1. Программный код функции **count\_general\_statistics\_percent** на языке программирования PHP:

function count\_general\_statistics\_percent($region) {

$result = array();

global $connect;

global $crime\_articles;

global $causes\_of\_crimes;

global $number\_of\_victims;

// -- Return arrow to start of query string result

mysqli\_data\_seek($crime\_articles, 0);

mysqli\_data\_seek($causes\_of\_crimes, 0);

mysqli\_data\_seek($number\_of\_victims, 0);

$total\_count = 0;

// -- Total sums:

while ($row = mysqli\_fetch\_assoc($crime\_articles)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

$total\_count += $row["importance\_of\_the\_statistical\_factor"];

}

}

while ($row = mysqli\_fetch\_assoc($causes\_of\_crimes)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

$total\_count += $row["importance\_of\_the\_statistical\_factor"];

}

}

while ($row = mysqli\_fetch\_assoc($number\_of\_victims)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

$total\_count += $row["importance\_of\_the\_statistical\_factor"];

}

}

// -- Return arrow to start of query string result

mysqli\_data\_seek($crime\_articles, 0);

mysqli\_data\_seek($causes\_of\_crimes, 0);

mysqli\_data\_seek($number\_of\_victims, 0);

// -- Count total values for every name of statistical factor of current region:

while ($row = mysqli\_fetch\_assoc($number\_of\_victims)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

if (isset($result["Общее количество потервпеших"])) {

$result["Общее количество потервпеших"] += $row["importance\_of\_the\_statistical\_factor"];

} else {

$result["Общее количество потервпеших"] = $row["importance\_of\_the\_statistical\_factor"];

}

}

}

while ($row = mysqli\_fetch\_assoc($causes\_of\_crimes)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

if (isset($result["Алкогольные, токсические, наркотические опьянения или лица, ранее совершавшие преступления"])) {

$result["Алкогольные, токсические, наркотические опьянения или лица, ранее совершавшие преступления"] += $row["importance\_of\_the\_statistical\_factor"];

} else {

$result["Алкогольные, токсические, наркотические опьянения или лица, ранее совершавшие преступления"] = $row["importance\_of\_the\_statistical\_factor"];

}

}

}

while ($row = mysqli\_fetch\_assoc($crime\_articles)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

if (isset($result["Общее число нарушений УК РФ"])) {

$result["Общее число нарушений УК РФ"] += $row["importance\_of\_the\_statistical\_factor"];

} else {

$result["Общее число нарушений УК РФ"] = $row["importance\_of\_the\_statistical\_factor"];

}

}

}

// -- Count percent values for every region:

foreach ($result as $key => $value) {

$result[$key] = round($value / $total\_count, 3) \* 100;

}

return $result;

}

1. Программный код функции **count\_quantitative\_values** на языке программирования PHP:

function count\_quantitative\_values($region, $query) {

$result = array();

global $connect;

mysqli\_data\_seek($query, 0);

// -- Create array of data [$key => region, $value => amount of the statistical factor]

while ($row = mysqli\_fetch\_assoc($query)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

$result[$row["name\_of\_the\_statistical\_factor"]] = $row["importance\_of\_the\_statistical\_factor"];

}

}

return $result;

}

1. Программный код функции **count\_percent\_values** на языке программирования PHP:

function count\_percent\_values($region, $query) {

$result = array();

global $connect;

// -- Return arrow to start of query string result

mysqli\_data\_seek($query, 0);

$total\_sum = 0;

// -- Count sum of all statistical factors for region

while ($row = mysqli\_fetch\_assoc($query)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

$total\_sum += $row["importance\_of\_the\_statistical\_factor"];

}

}

// -- Return arrow to start of query string result

mysqli\_data\_seek($query, 0);

// -- Create array of data [$key => region, $value => percent of the statistical factor]

while ($row = mysqli\_fetch\_assoc($query)) {

if (preg\_replace('/\s+/', '', $row["subject"]) == $region) {

$result[$row["name\_of\_the\_statistical\_factor"]] = round($row["importance\_of\_the\_statistical\_factor"] / $total\_sum, 3) \* 100;

}

}

return $result;

}

1. Программный код функции **count\_dispersion** на языке программирования PHP:

function count\_dispersion($query) {

$result = array();

global $connect;

global $regions;

// -- Return arrow to start of query string result

mysqli\_data\_seek($query, 0);

$total\_sum = 0;

$total\_count = 0;

$total\_sum\_of\_the\_statistical\_factors = array();

$average\_of\_the\_statistical\_factors = array();

$dispersions = array();

// -- Count total sum and count of the statistical factor

while ($row = mysqli\_fetch\_assoc($query)) {

if ($row["subject"] != "Всего по России") {

if (isset($total\_sum\_of\_the\_statistical\_factors[$row["name\_of\_the\_statistical\_factor"]])) {

$total\_sum\_of\_the\_statistical\_factors[$row["name\_of\_the\_statistical\_factor"]] += $row["importance\_of\_the\_statistical\_factor"];

} else {

$total\_sum\_of\_the\_statistical\_factors[$row["name\_of\_the\_statistical\_factor"]] = $row["importance\_of\_the\_statistical\_factor"];

}

}

}

// -- Count average for each statistical factor

foreach ($total\_sum\_of\_the\_statistical\_factors as $key => $value) {

$average\_of\_the\_statistical\_factors[$key] = round($total\_sum\_of\_the\_statistical\_factors[$key] / (count($regions) - 1), 2);

}

// -- Return arrow to start of query string result

mysqli\_data\_seek($query, 0);

// -- Count dispersion for each statistical factor

$numerators = array();

$result = 0;

while ($row = mysqli\_fetch\_assoc($query)) {

if ($row["subject"] != "Всего по России") {

foreach ($average\_of\_the\_statistical\_factors as $key => $value) {

if ($key == $row["name\_of\_the\_statistical\_factor"]) {

if (isset($numerators[$key])) {

$numerators[$key] += pow($row["importance\_of\_the\_statistical\_factor"] - $value, 2);

$result = sqrt($numerators[$key] / (count($regions) - 1));

$dispersions[$key] = round($result, 2);

} else {

$numerators[$key] = pow($row["importance\_of\_the\_statistical\_factor"] - $value, 2);

}

}

}

}

}

return $dispersions;

}