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: *«Разработка веб-приложения для повышения информированности о нарушениях 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 повысить информационность пользователя о статистике совершаемых дорожных происшествий в регионах России, а также предоставить 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, отображающего статистику нарушений дорожно-транспортных происшествий в регионах России в целях повышения информационности пользователя, 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 выгрузки 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 происшествия [1] и безопасность дорожного движения [2]. Исходные датасеты были очищены от лишних символов «пробел», а также была отредактирована первая строчка под названия атрибутов для выгрузки 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;
2. regional\_statistics.php;
3. statistics\_of\_victims.php;
4. tips\_and\_recommendations.php;
5. tips.php;
6. dbconnect.php.

index.php – главная страница приложения. Содержит краткое описание проекта, меню для выбора интересующей статистики или рекомендаций, а также ссылки на использованные открытые данные.

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.

statistics\_of\_victims.php – станица, 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.

tips\_and\_recommendations.php – страница, содержащая меню для выбора типа пользователя, а также предоставляющая советы водителям и пешеходам.

tips.php – компонент, содержащий статический текст советов и рекомендаций в массиве.

dbconnect.php – компонент, содержащий код подключения к базе данных.

Исходные открытые данные были занесены в две независимые сущности, получившие названия региональная статистика и статистика пострадавших. Реляционная модель получившейся базы данных представлена на рисунке 1.

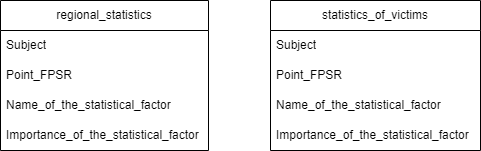


Рисунок 1 – Реляционная модель

Диаграмма компонентов приложения, написанная на языке диаграмм UML, описывает взаимодействие между всеми компонентами веб-приложения и представлена на рисунке 2.

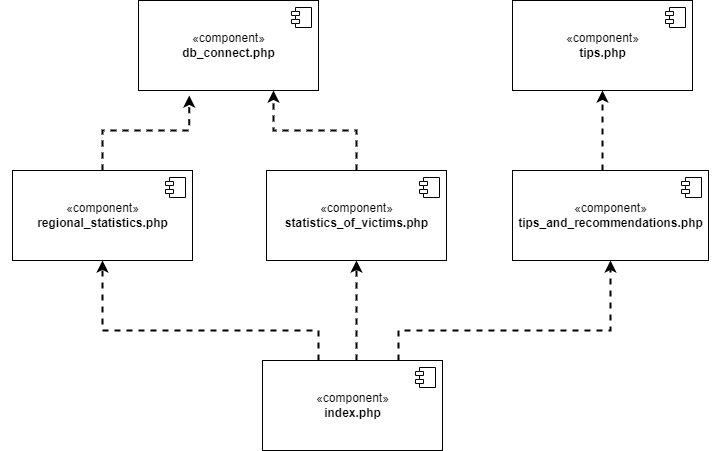


Рисунок 2 – Диаграмма компонентов

**3 Реализация приложения**

Разработка приложения началась с нахождения необходимых вспомогательных библиотек для построения графиков. Для создания динамический графиков была выбрана библиотека AnyChart.js [3]. Далее были сверстаны макеты для будущих PHP-компонентов, а также описаны стили для каждого из них в отдельной директории. Верстка выполнена на основе CSS flexbox (Flexible Box Layout Module) – системы компоновки элементов. Первоначальная настройка для диаграмм была задана из документации используемой библиотеки. Для приложения был изменен цвет столбцов диаграммы и длина подписей вертикальной оси. 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 формата csv 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 array\_sort, который после использовался для вывода содержимого на страницу с помощью цикла. Элементы массива являются ссылками, передающие в атрибут region название региона. Ниже списка располагаются все доступны графики в виде горизонтальных столбчатых диаграмм, созданные с помощью тега canvas, имеющий атрибут id для доступа к настройке конкретного графика. Настройки для графиков описаны в самом низу документа, в теге script. Настройки определяют размеры, атрибуты, цветовую палитру диаграммы.

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

1. преобразование исходных открытых данных в необходимый для графика формат происходит с помощью SELECT-запроса нужного датасета, который итерируется с помощью цикла while и функции mysqli\_fetch\_assoc для преобразования результата запроса в понятный для PHP формат. Результат на выходе является массивом, формат которого соответствует формуле (1) (см. приложение А листинг – 1 и листинг – 4).

|  |  |  |
| --- | --- | --- |
|  | Array ([key, value], [key, value] …) |  |

1. преобразование исходных данных в процентное соотношение происходит, согласно формуле (2), где s – сумма всех показателей, n – их количество. Первым циклом определяется полное название региона со всеми пробелами для подстановки в условие запроса. Вторым циклом создается массив с данными, название которых содержит слово «всего». Финальный цикл итерируется по результату запроса с условием названия текущего выбранного региона, в котором каждая строка данных делится на соответствующее значение со словом «всего» из массива, ключи которого начинаются с такого же слова, что и итерируемый объект, а также имеют слово «всего». После чего результат умножается на 100. Результат на выходе является массивом, формат которого соответствует формуле (1) (см. приложение А листинг – 2).

|  |  |  |
| --- | --- | --- |
|  | \* 100 |  |

1. вычисление стандартного отклонения для каждого из показателей необходимого датасета происходит, согласно формуле (3), где D – дисперсия, x – значение показателя, – среднее значение, n – количество показателей. Первый цикл проводит итерацию по результату SELECT-запроса необходимого датасета для вычисления суммы по каждому статистическому фактору со всех регионов, кроме объекта с названием «Всего по России». Второй цикл определяет массив средних значений в каждом статистической факторе. Следующий цикл высчитывает массив стандартных отклонений для каждого атрибута. Финальный цикл преобразует результат данных в необходимый формат. Результат на выходе является массивом, формат которого соответствует формуле (1) (см. приложение А листинг – 3 и листинг – 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.

Завершающим этапом разработки приложения предстала цель размещения готового продукта на хостинг московского политехнического университета в целях всеобщего обозрения.

**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 главной страницы изображен на рисунке 3.

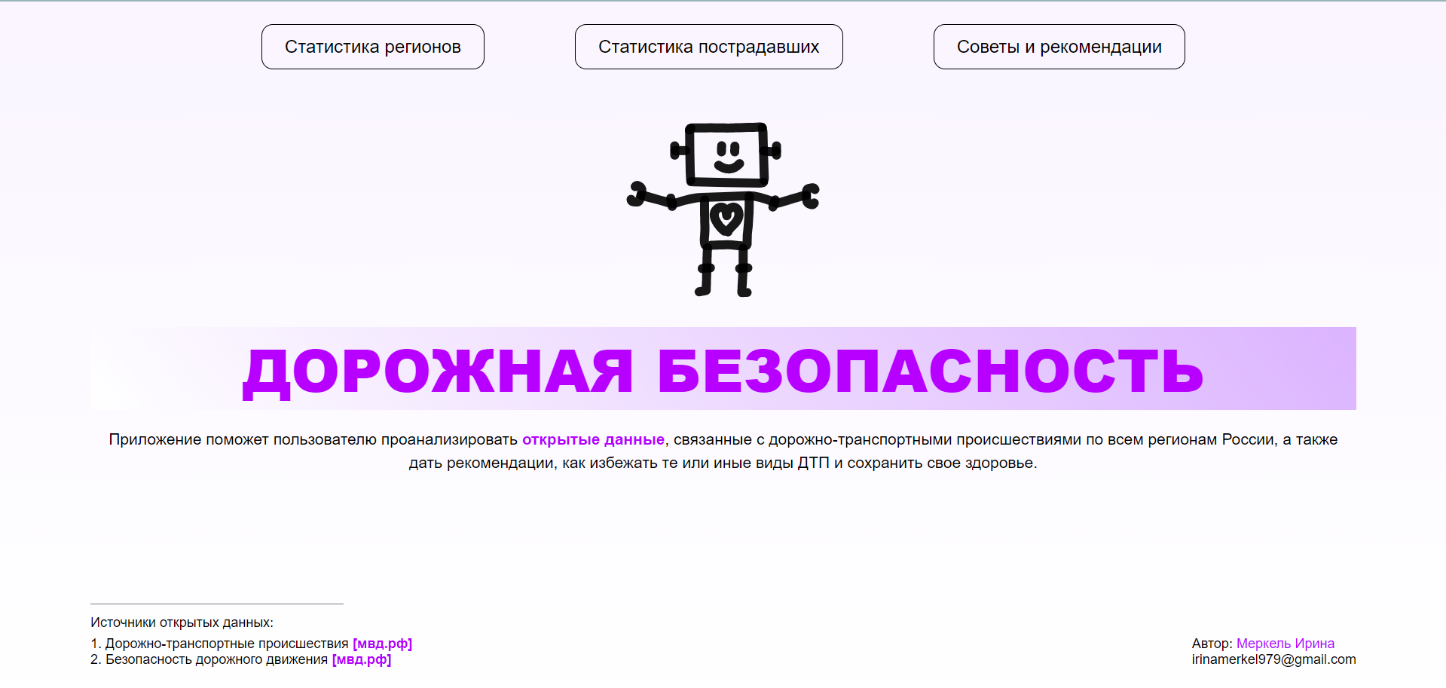


Рисунок 3 – Интерфейс главной страницы

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 для страницы со статистикой региона представлен на рисунке 4.

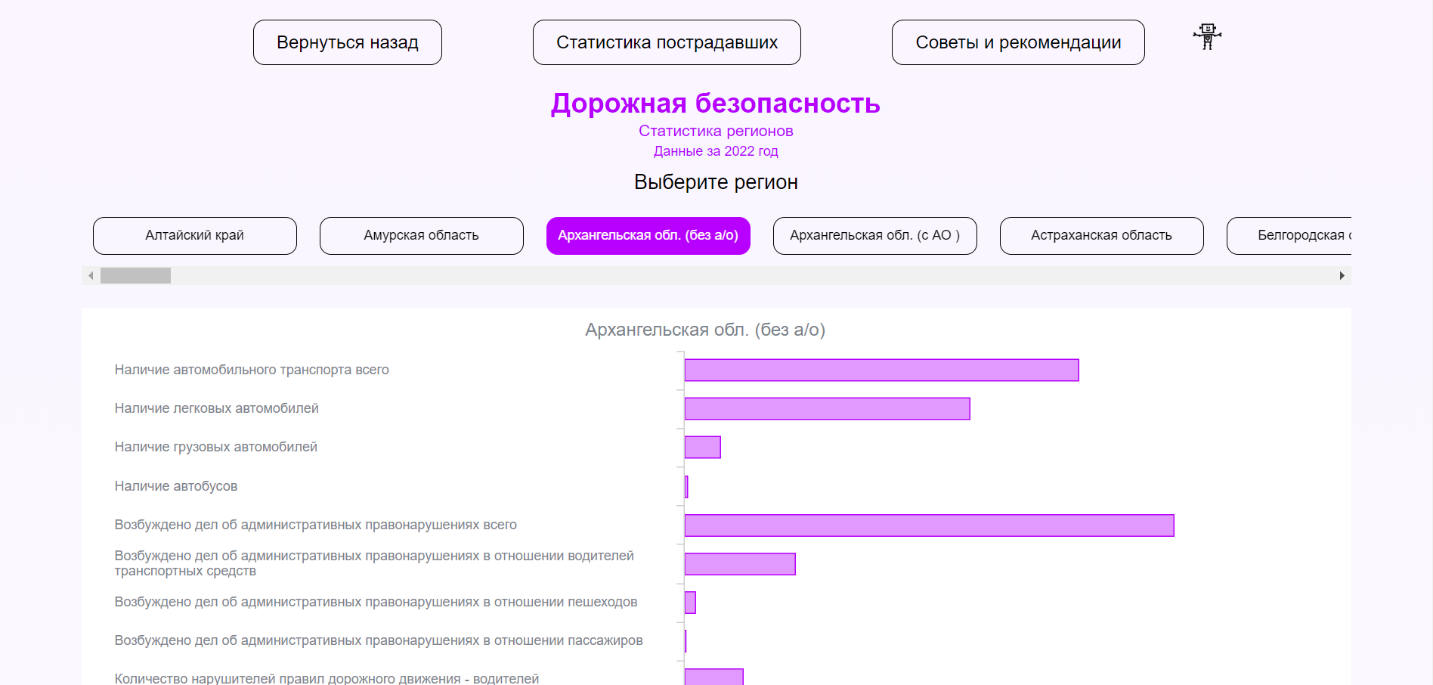


Рисунок 4 – Интерфейс для страницы со статистикой региона

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 для страницы со статистикой пострадавших представлен на рисунке 5.

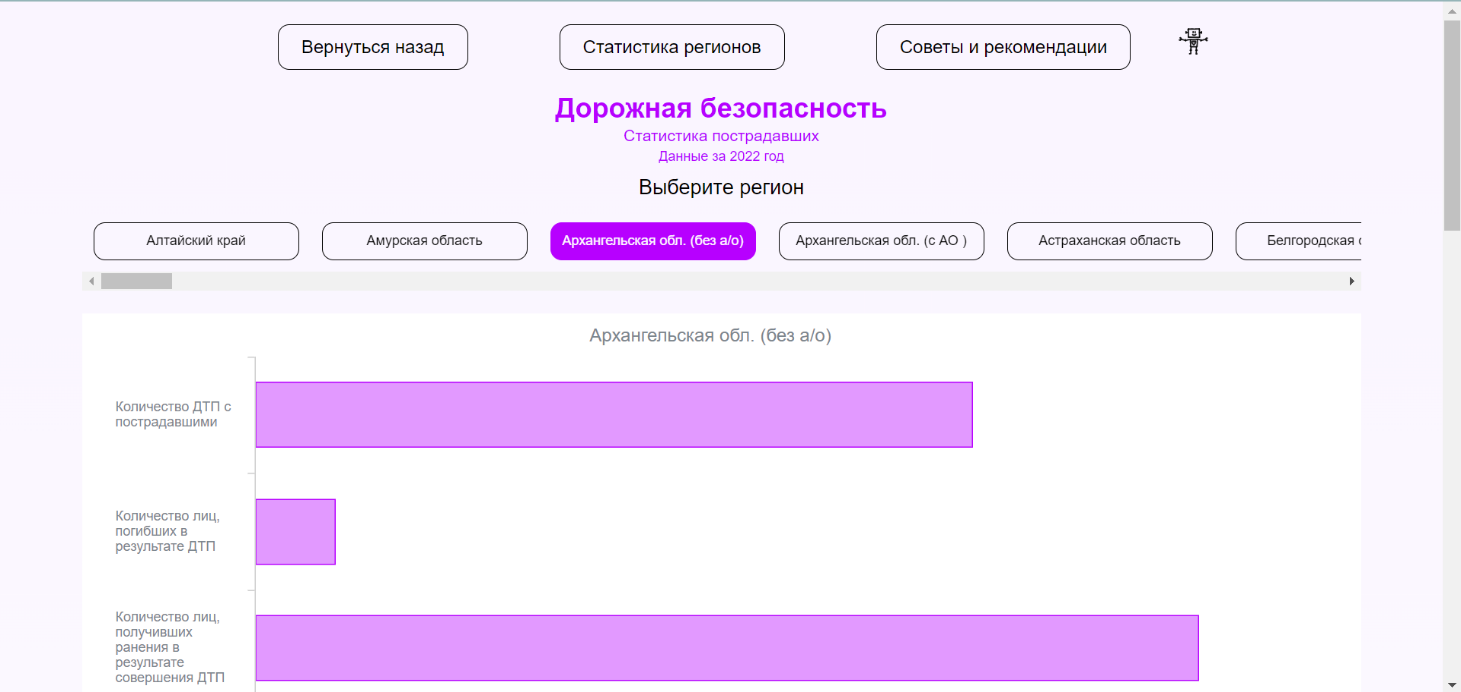


Рисунок 5 – Интерфейс для страницы со статистикой пострадавших

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 «Вернуться назад», «Статистика регионов» и «Статистика пострадавших» пользователь может переходить на соответствующие страницы. Интерфейс для страницы с советами и рекомендациями представлен на рисунке 6.

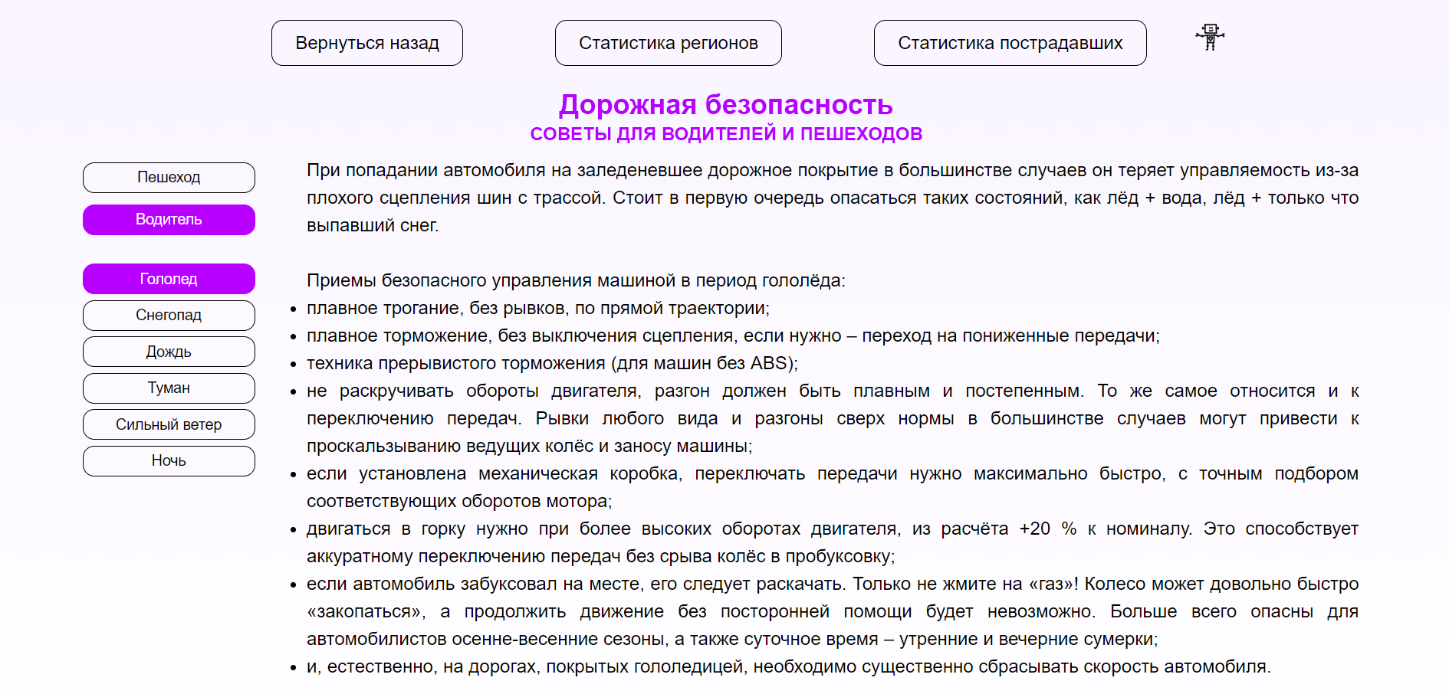


Рисунок 6 – Интерфейс для страницы с советами и рекомендациями

**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 путем добавления разнообразных диаграмм, новых советов и рекомендаций и возможностью регистрироваться в приложении.

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. Дорожно-транспортные происшествия. [Электронный ресурс]. URL: <https://xn--b1aew.xn--p1ai/opendata/7727739372-MVD_GIAC_3.1> (дата обращения: 10.01.23).
2. Безопасность дорожного движения. [Электронный ресурс]. URL: <https://xn--b1aew.xn--p1ai/opendata/7727739372-MVDGIAC32> (дата обращения: 10.01.23).
3. Галерея Графиков И Диаграмм. [Электронный ресурс]. URL: <https://www.anychart.com/ru/products/anychart/gallery/> (дата обращения: 10.01.23).
4. 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. Безопасность пешеходов. [Электронный ресурс]. URL: <https://xn--90adear.xn--p1ai/mens/pravo-peshehoda> (дата обращения: 12.01.23).

**Приложение А**

*(справочное)*

1. Листинг 1 – Выгрузка исходных открытых данных в динамический график на странице статистики 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); |

1. Листинг 2 – Преобразование исходных открытых данных датасета для страницы статистики регионов в процентном соотношении.

|  |
| --- |
| $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); |

1. Листинг 3 - Вычисление среднего отклонения для каждого показателя в датасете на странице статистики 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); |

1. Листинг 4 – Выгрузка исходных открытых данных в динамический график на странице статистики 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); |

1. Листинг 5 – Вычисление стандартного отклонения показателей в датасете статистики 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); |