

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
МОРСКОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
Кафедра вычислительной техники  
и информационных технологий

**Курсовая работа**  
по дисциплине  
«Разработка WEB приложения»

**ИСПОЛНИТЕЛЬ**

Студент группы 20390

**Ренжин А. Ю.**

**РУКОВОДИТЕЛЬ**

Доцент КВТ и ИТ

**В.А.Семенова-Тян-Шанская**

Санкт-Петербург

2021

## Оглавление

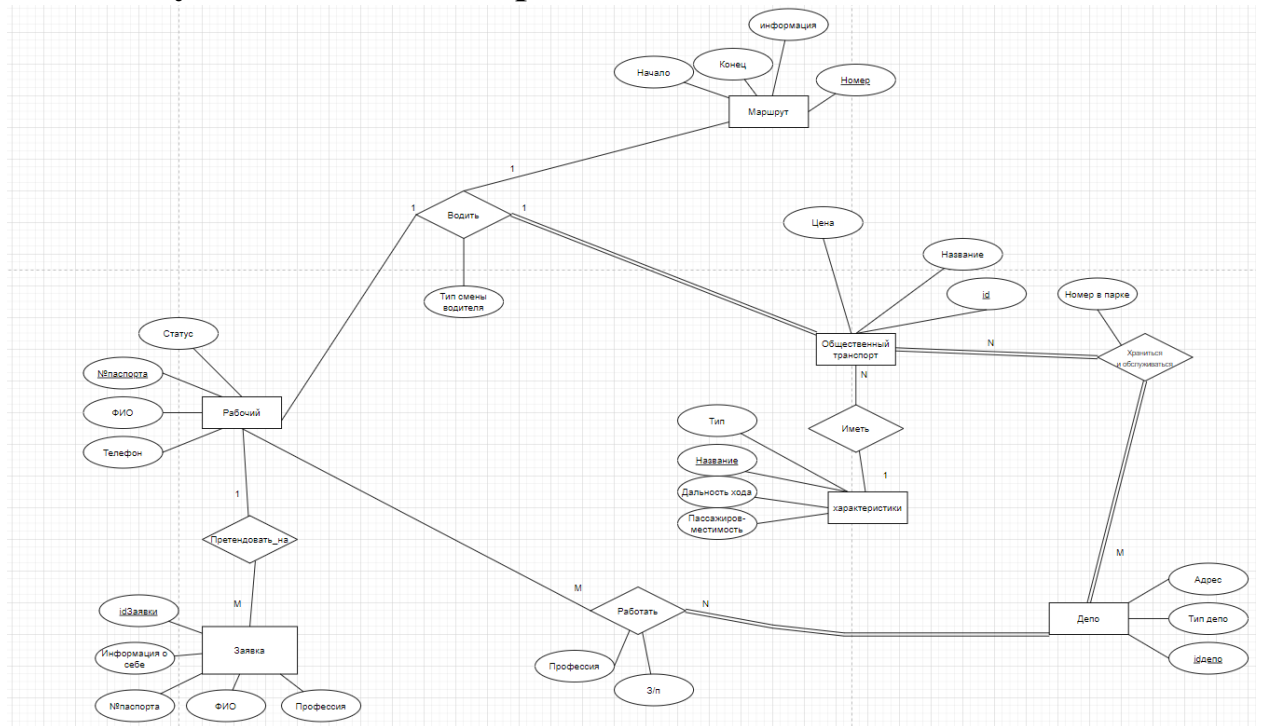
Разработка приложений на ASP.NET с помощью технологии MVC .....	3
Концептуальная модель предметной области .....	4
Логическая модель предметной области .....	4
Создание модели .....	5
Каскадное удаление .....	6
Триггер для автоматической подстановки номера .....	6
Схема данных .....	8
Иерархия страниц.....	9
Описание работы приложения .....	9
Главная страница.....	9
Использование мастер-страницы.....	10
Пояснение .....	11
Вход.....	11
Заявка на работу .....	11
Информация о предприятии.....	12
Транспорт на маршруте .....	12
Главная страница работника .....	13
Профессия работника .....	14
Список сотрудников .....	14
Код страницы.....	14
Пояснение .....	15
Метод страницы .....	15
Главная страница менеджера .....	16
«Рабочее место» менеджера.....	16
Страница администратора.....	19
Обработка SELECT-запроса.....	19
Обработка запроса на Вставку, Изменение, Удаление в таблицу .....	20
Обработка ошибки .....	21
Код страницы.....	22
Пояснения .....	22
Код мастер-страницы.....	22
Метод страницы .....	23

## Разработка приложений на ASP.NET с помощью технологии MVC

Данное приложение разработано с применением языка C# и паттерна MVC. Паттерн MVC(Model-View-Controller) – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

- Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.
- Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.
- Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

## Концептуальная модель предметной области



## Логическая модель предметной области

Сотрудник (№Паспорта, ФИО, Телефон, Адрес, статусНаСайте)

ЗаявкаНаРаботу(№Заявки, №Паспорта, ФИО, Профессия, Телефон, ИнформацияОСебе)

Депозит(idДепозита, Адрес, Тип)

Техника(idТехники, цена, название)

Характеристики\_техники(название, запас хода, вместимость, тип)

Маршрут(номер, старт, финиш, информацияОМаршруте)

СотрудникиВДепозите(idДепозита, №Паспорта, з/п, профессия)

СотрудникиНаМаршруте(номерМаршрута, idТехники, №Паспорта, типСмены)

ТехникаВДепозите(номерВДепозита, idТехники, idДепозита)

## Создание модели

```
create database autopark
go
```

```
create table employer
(
    id_passport INT NOT NULL,
    name_emp varchar (50) NOT NULL,
    adress varchar (50) NOT NULL,
    telephone varchar (10) NOT NULL,
    status_ varchar (50) CHECK (type_dep IN ('manager', 'worker', 'dispatcher', 'admin')),
    login_ varchar(20) NOT NULL UNIQUE,
    password_ VARCHAR(20) NOT NULL,
    primary key (id_passport))
go
```

```
create table job_apply
(
    id_apply INT NOT NULL IDENTITY(1,1),
    id_passport INT NOT NULL,
    name_j_apply varchar (50) NOT NULL,
    name_prof varchar(50) NOT NULL,
    telephone varchar (10) NOT NULL,
    info_j_apply varchar (80),
    primary key (id_apply))
go
```

```
create table depot
(
    id_dep varchar (5) NOT NULL,
    adress varchar (50) NOT NULL,
    type_dep varchar (10) CHECK (type_dep IN ('Tram', 'Electro', 'Trol')),
    primary key (id_dep))
go
```

```
create table tech_prop
(
    name_tec varchar (50) NOT NULL,
    autonomy int NOT NULL,
    capacity int NOT NULL,
    tec_type varchar(10) NOT NULL CHECK (tec_type IN ('Tram', 'Electro', 'Trol')),
    primary key (name_tec))
go
```

```
create table technic
(
    id_tec int NOT NULL IDENTITY(1,1),
    price int NOT NULL,
    name_tec varchar (50) NOT NULL,
    FOREIGN KEY (name_tec) REFERENCES tech_prop(name_tec),
    primary key (id_tec))
go
```

```
create table city_route
(
    number varchar (5) NOT NULL,
    start_ varchar (50) NOT NULL,
    finish_ varchar (50) NOT NULL,
    info_c_route varchar(80),
    primary key (number))
go
```

```

create table emp_tec_route
(
number varchar (5) NOT NULL,
id_tec int NOT NULL,
id_passport int NOT NULL,
type_e_t varchar(10) NOT NULL CHECK (type_e_t IN ('morning', 'even'))
)
go

```

```

create table emp_dep
(
id_dep varchar(5) NOT NULL,
id_passport int NOT NULL,
salary int NOT NULL,
profession varchar(30) NOT NULL,
)
go

```

```

create table tec_dep
(
number_in_dep int NOT NULL,
id_tec int NOT NULL,
id_dep varchar (5) NOT NULL
)
go

```

## Каскадное удаление

```

ALTER TABLE emp_dep
ADD CONSTRAINT fk_id_dep FOREIGN KEY (id_dep) REFERENCES depot(id_dep) ON DELETE CASCADE

```

```

ALTER TABLE emp_dep
ADD CONSTRAINT fk_id_passport FOREIGN KEY (id_passport) REFERENCES employer(id_passport)
ON DELETE CASCADE;

```

```

ALTER TABLE emp_tec_route ADD CONSTRAINT fk_id_tec_r FOREIGN KEY (id_tec) REFERENCES
technic(id_tec) ON DELETE CASCADE;

```

```

ALTER TABLE emp_tec_route ADD CONSTRAINT fk_id_passport_r FOREIGN KEY (id_passport)
REFERENCES employer(id_passport) ON DELETE CASCADE;

```

```

ALTER TABLE emp_tec_route ADD CONSTRAINT fk_number_r FOREIGN KEY (number) REFERENCES
city_route(number) ON DELETE CASCADE;

```

```

ALTER TABLE tec_dep ADD CONSTRAINT fk_id_tec_td FOREIGN KEY (id_tec) REFERENCES
technic(id_tec) ON DELETE CASCADE

```

```

ALTER TABLE tec_dep ADD CONSTRAINT fk_id_id_dep_td FOREIGN KEY (id_dep) REFERENCES
depot(id_dep) ON DELETE CASCADE ;

```

## Триггер для автоматической подстановки номера

```

ALTER TRIGGER [dbo].[number_in_dep]
ON [dbo].[tec_dep]
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @num int

    SELECT @num = (SELECT coalesce(MAX(number_in_dep),0) FROM tec_dep WHERE id_dep =
(SELECT id_dep FROM inserted) )+1

```

```
INSERT INTO tec_dep (number_in_dep, id_tec, id_dep)
SELECT @num, inserted.id_tec, inserted.id_dep
FROM inserted
```

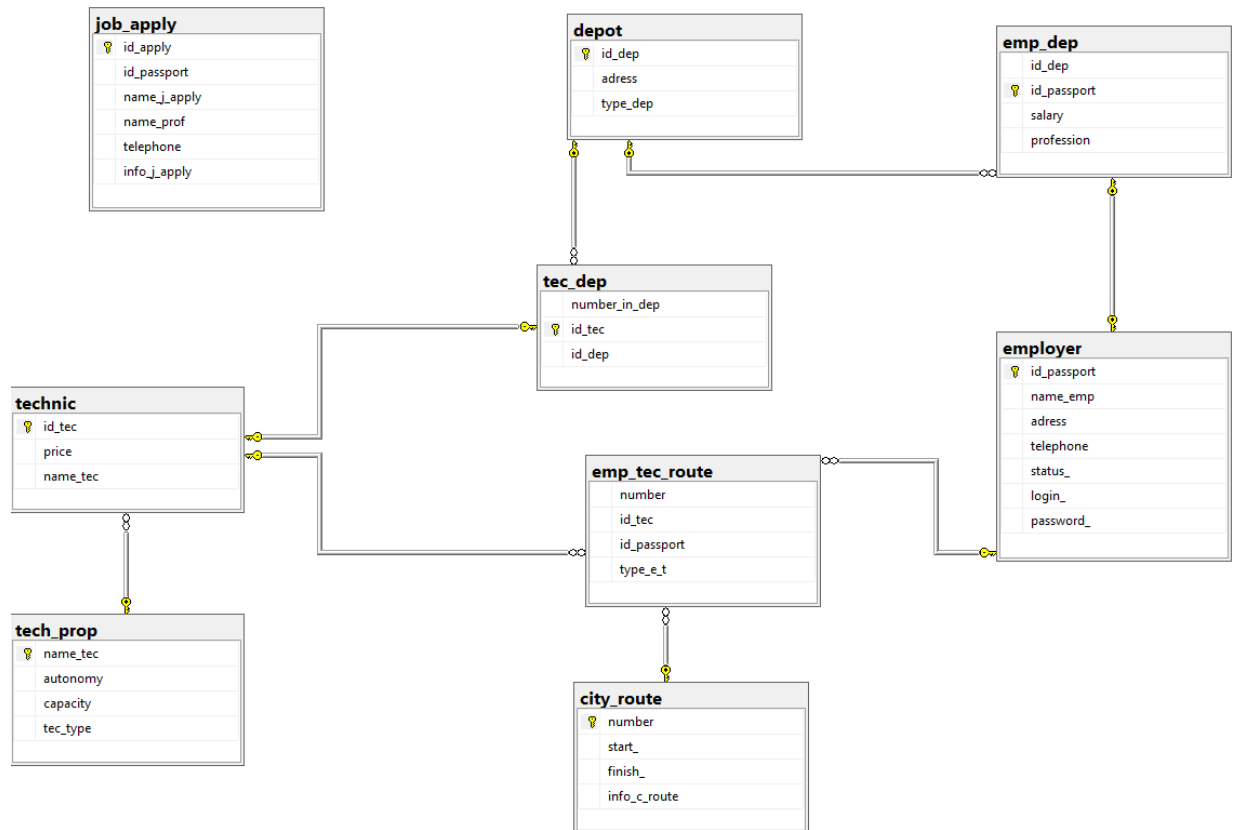
END

Пример работы триггера:

### Таблица транспорта в депо

idТехники	idДепо	Номер техники в этом депо
1	A	1
2	A	2
3	B	1
4	C	1

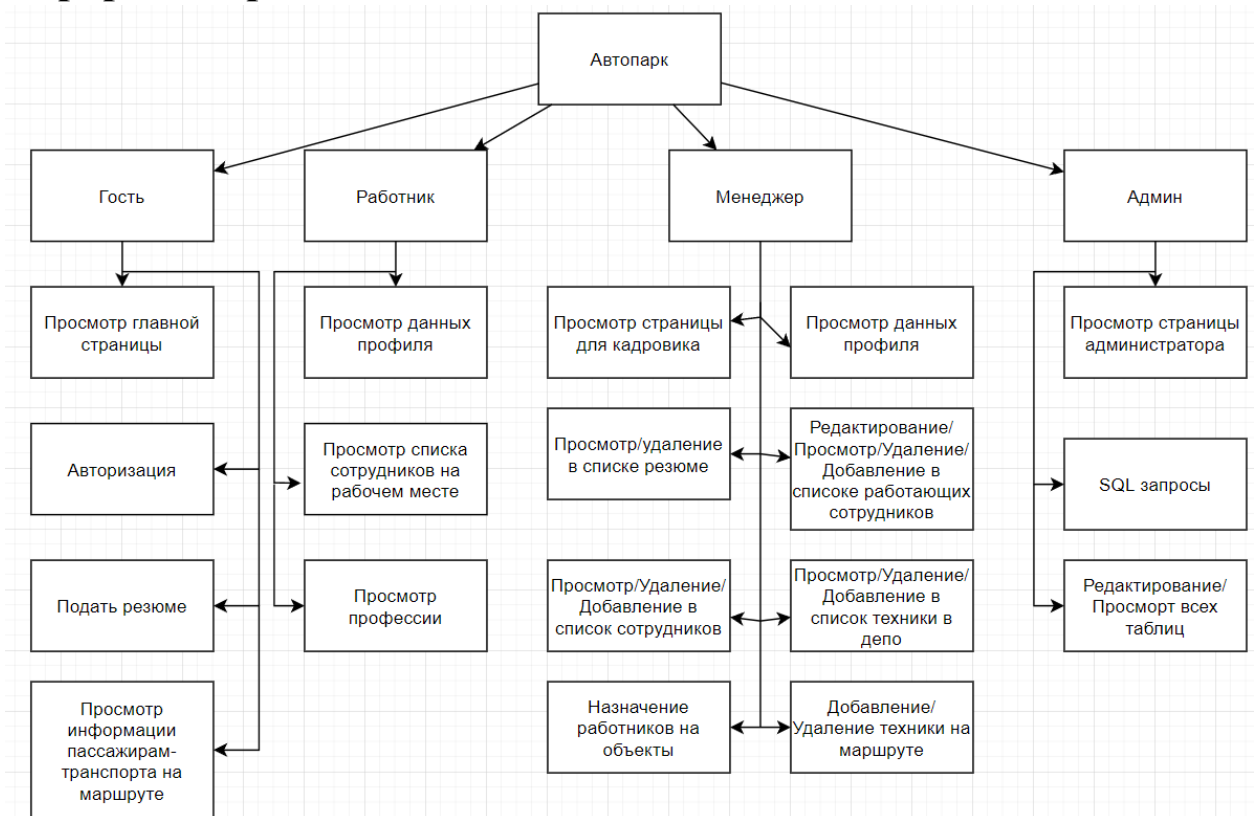
## Схема данных



Заявки на работу рассматривает менеджер поэтому таблица ЗаявкаНаРаботу никак не связана с остальной системой



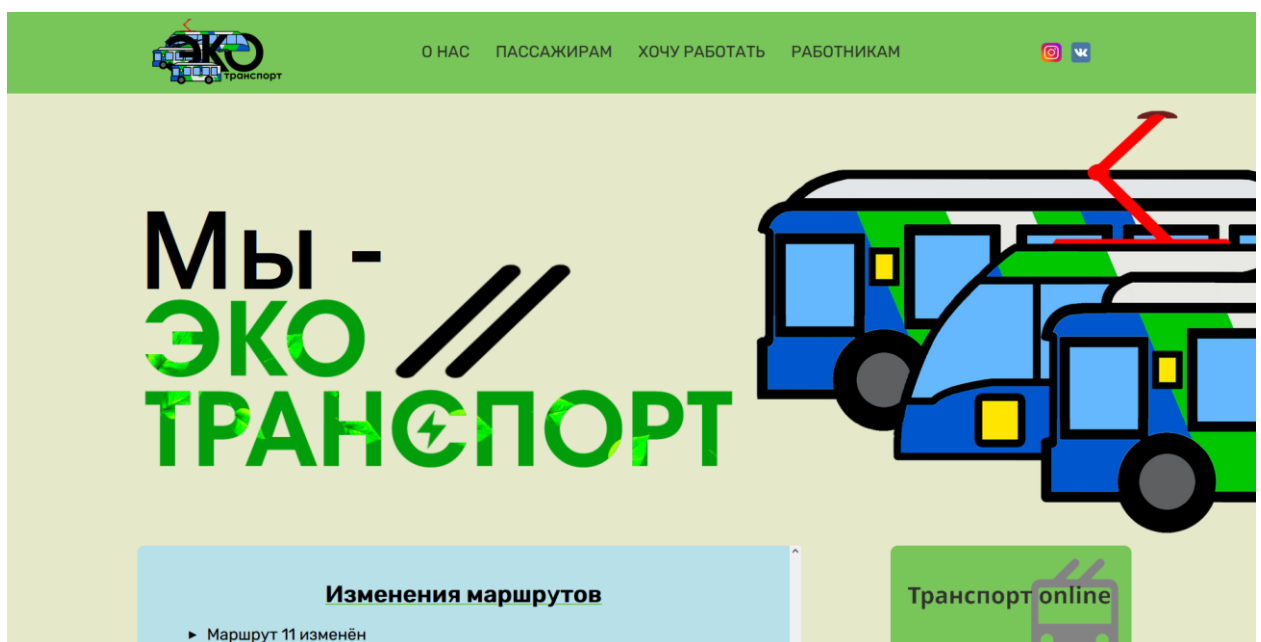
## Иерархия страниц

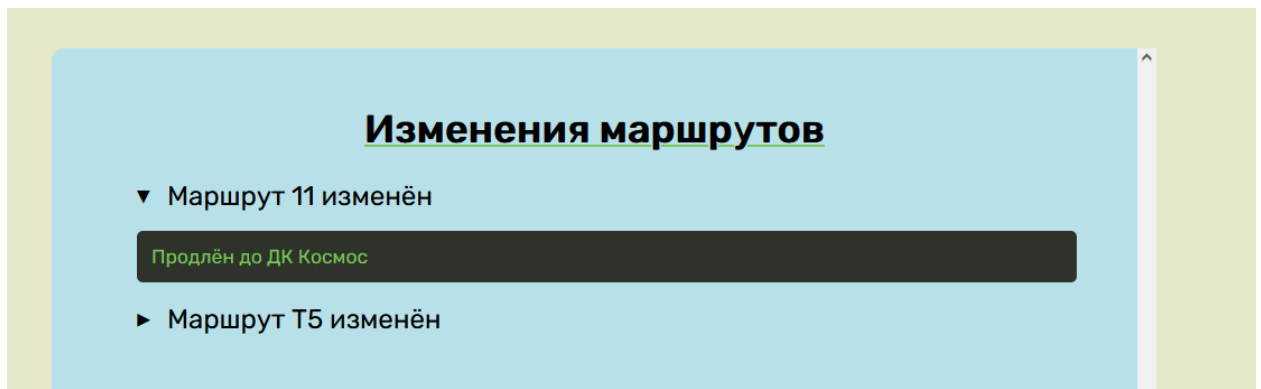


## Описание работы приложения

### Главная страница

При запуске приложения попадаем на главную страницу для не авторизованного пользователя, на котором мы можем посмотреть информацию о предприятии, подать заявку на работу, посмотреть актуальную информацию по транспорте на маршруте, перейти на страницу авторизации, а также можно увидеть изменения на некоторых маршрутах.





## Использование мастер-страницы

Все последующие представления использует данную мастер страницу – шаблон, который прописывает общие элементы. В моем случае это данные «шапки» и нижнего колонтитула сайта.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ЭкоТранспорт</title>
  <link rel="stylesheet" href="/Content/Style.css">
</head>

<body style="overflow-x:hidden;">

<header class="header">
  <div class="header_selection">
    <div class="logo" style="float: left; margin-top:6px; margin-left:215px;">

      <a href="~/Home/Index"></a>
    </div>
  </div>

  <nav class="nav">
    <a class="nav__link" href="~/Home/Inform">0 нас</a>

    <a class="nav__link" href="~/Home/Passenger">Пассажирам</a>
    <a class="nav__link" href="~/Home/Application">Хочу работать</a>

    <a class="nav__link" href="~/Home/Authorization">Работникам</a>
  </nav>

  <div class="header_selection" style="float: right; margin-top:6px; margin-right:215px;">
    <a class="social__link" href="https://instagram.com">
      
    </a>
    <a class="social__link" href="https://vk.com">
      
    </a>
  </div>
</header>
<div class="main">
  @RenderBody()
</div>
```

```
<footer class="footer">
  Все права защищены
  <br>
  © 2021 Ренжин Андрей
</footer>
</body>
</html>
```

## Пояснение

**@RenderBody()**

- на это место другие представления, которые используют эту мастер-страницу, будут подставлять свое содержимое.

## Вход

При нажатии на кнопку «Работникам» попадаем на страницу авторизации, с которой при вводе логина-пароля попадаем на соответствующие главные страницы авторизованного пользователя: «работника», «менеджера», «админа»



## Заявка на работу

При нажатии на кнопку «Хочу работать» попадаем на страницу с формой, которую пользователь может заполнить чтобы подать заявку для работы в компании.



## Заявка на работу

ФИО:

Паспорт:

Профессия:

Номер телефона:

Опыт работы кратко:

## Информация о предприятии

При нажатии на кнопку «О нас» попадаем на страницу с информацией.



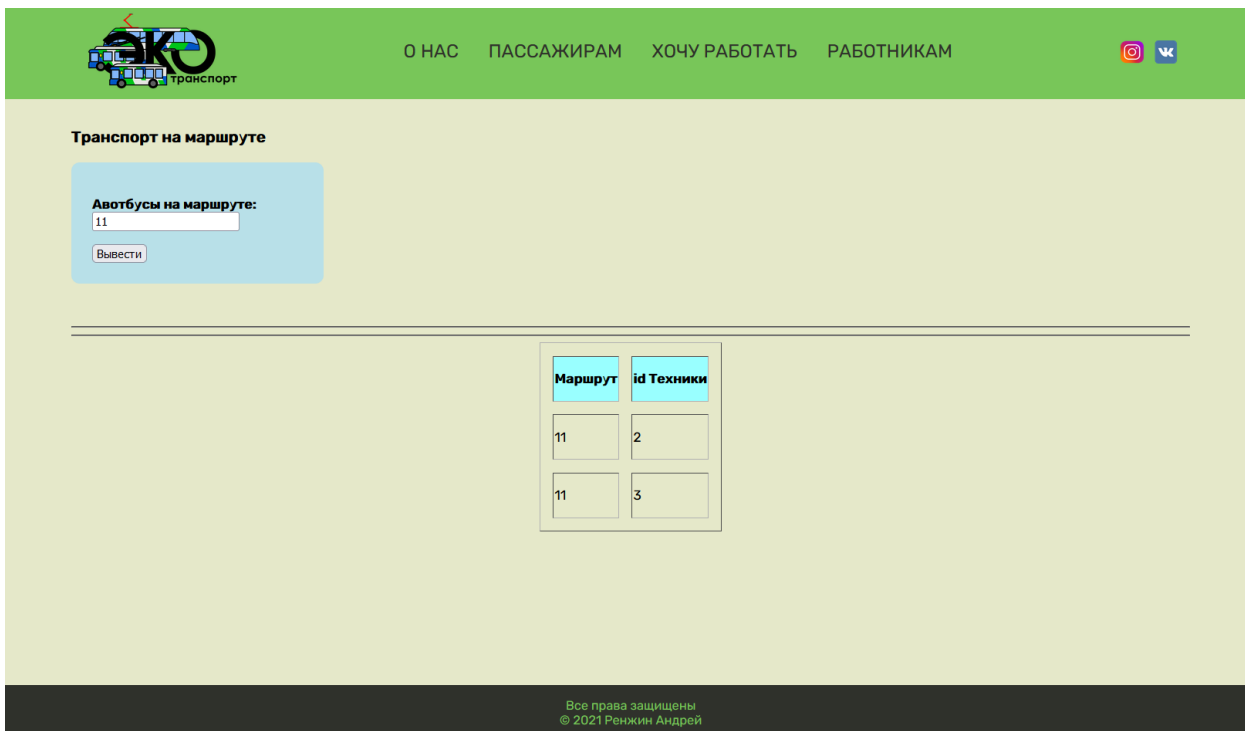
## Наша фирма

Занимаемся пассажирскими перевозками в городе Киров и за его пределами.



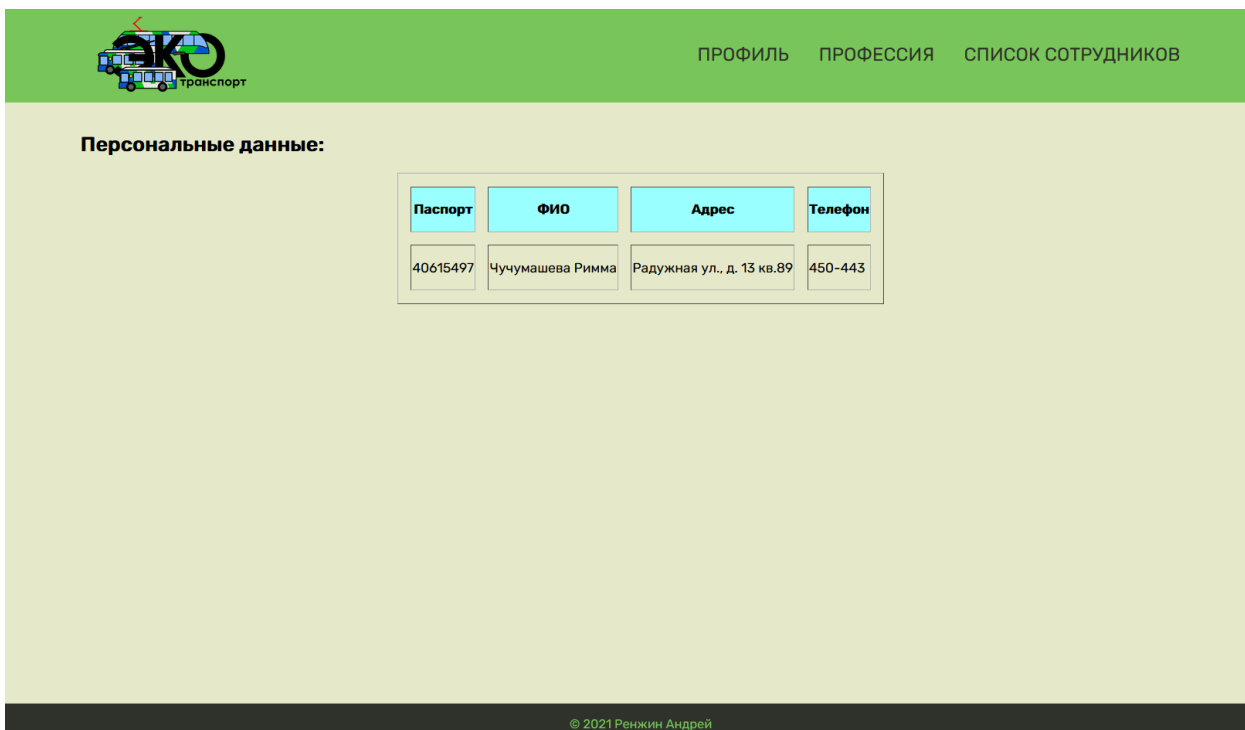
## Транспорт на маршруте

Пользователь вводит в форму номер маршрута и по его запросу выводится транспорт, который сейчас работает на маршруте.




## Главная страница работника

После входа как «работник» попадаем на страницу, на которой выведена личная информация, с этой страницы можно попасть на страницу с информацией о своем месте работы (профессии) и к списку сотрудников.



## Профессия работника



ПРОФИЛЬ   ПРОФЕССИЯ   СПИСОК СОТРУДНИКОВ


Данные профессии:

Паспорт	Место работы	З/п	Профессия
40615497	Депо #А	25000	Электрик

© 2021 Ренжин Андрей

## Список сотрудников

Данная таблица образована путем соединения таблицы сотрудников и таблицы профессий сотрудников.



ПРОФИЛЬ   ПРОФЕССИЯ   СПИСОК СОТРУДНИКОВ

Список сотрудников

Имя	Адресс	Телефон	Место работы	Профессия
Чучумашева Римма	Радужная ул., д. 13 кв.89	450-443	Депо # А	Электрик
Липин Александр	Калинина ул., д. 15 кв.156	560-257	Депо # С	Электрик
Богомазов Афанасий	Колхозный пер., д. 14 кв.56	440-439	Депо # В	Водитель

© 2021 Ренжин Андрей

## Код страницы

```
@{  
    Layout = "~/Views/Shared/_LayoutManager.cshtml";  
}
```

```

<h3>Список сотрудников</h3>
<table border="1" align="center" cellspacing="15" >
  <tr bgcolor="#99FFFF">
    <th>
      <p>Имя</p>
    </th>
    <th>
      <p>Адрес</p>
    </th>
    <th>
      <p>Телефон</p>
    </th>
    <th>
      <p>Место работы</p>
    </th>
    <th>
      <p>Профессия</p>
    </th>
  </tr>
  @foreach (var d in ViewBag.members)
  {
    <tr>
      <td><p>@d.emp.name_emp</p></td>
      <td><p>@d.emp.adress</p></td>
      <td><p>@d.emp.telephone</p></td>
      <td><p>Деп # @d.ed.id_dep</p></td>
      <td><p>@d.ed.profession</p></td>
    </tr>
  }
</table>

```

#### Пояснение

```

@foreach (var d in ViewBag.members)
{
  <tr>
    <td><p>@d.emp.name_emp</p></td>
    <td><p>@d.emp.adress</p></td>
    <td><p>@d.emp.telephone</p></td>
    <td><p>Деп # @d.ed.id_dep</p></td>
    <td><p>@d.ed.profession</p></td>
  </tr>
}

```

- заполняет таблицу содержанием из ViewBag

```

@{
  Layout = "~/Views/Shared/_LayoutManager.cshtml";
}

```

- использование мастер-страницы на которой можно определить общие для всех остальных представлений меню, а также подключить общие стили и скрипты.

#### Метод страницы

```

// GET: Work/Members
public ActionResult Members() //метод отображение таблицы сотрудников
{
  autoparkEntities dbWork = new autoparkEntities(); //переменная базы данных
  var users = from e in dbWork.employer
    join ed in dbWork.emp_dep on e.id_passport equals ed.id_passport
    select new ViewModel //использование вспомогательного класса
  {

```

```

        emp = e,
        ed = ed
    }; //объединение таблицы работников и работников в депо для
получения разной информации о работнике
    ViewBag.members = users; //передача во ViewBag получившейся таблицы
    return View(); //возвращение на представление для отображения списка
работников
}

public class ViewModel //вспомогательный класс, хранящий объекты базы данных
{
    public employer emp { get; set; }
    public emp_dep ed { get; set; }
}

```

## Главная страница менеджера

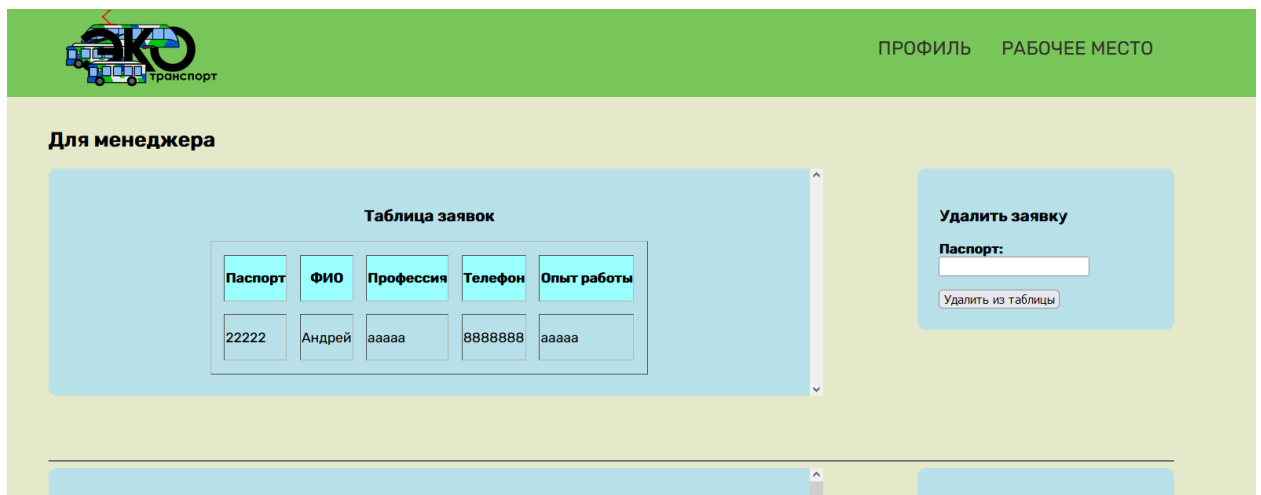
Здесь, как и на странице работника, на главной странице содержится информация с личной информации менеджера, с этой страницы можно попасть на «рабочее место»



Паспорт	ФИО	Адрес	Телефон
40151782	Зуйков Яков	Майская ул., д. 19 кв.217	440-656

## «Рабочее место» менеджера

На этой странице много форм для работы с различными таблицами: через них можно добавлять, удалять и изменять строки в таблицах.



Паспорт	ФИО	Профессия	Телефон	Опыт работы
22222	Андрей	ааааа	8888888	ааааа

**Удалить заявку**

Паспорт:



Таблица работников

Паспорт	ФИО	Адрес	Телефон	Роль	Логин	Пароль
12334567	Я	Стачек пр., д. 111	228-228	admin	admin	1234
40151782	Зуйков Яков	Майская ул., д. 19 кв.217	440-656	manager	yy-201	JDsa9S
40615497	Чучумашева Римма	Радужная ул., д. 13 кв.89	450-443	worker	Erellici	oIWoeQ
40973395	Липин Александр	Калинина ул., д. 15 кв.156	560-257	worker	Naria	xlO91V
41372958	Лагутова Марьяна	Школьная ул., д. 15 кв.103	120-157	manager	Huongar	1jsMhF
41419035	Петров Максим	Интернациональная ул., д. 17 кв.60	690-255	worker	Jesi	RerhpR
41467212	Оропай Александра	Южная ул., д. 19 кв.24	630-357	worker	Zanaki	4Mvplg

Добавить работника

Паспорт:

ФИО:

Адрес:

Телефон:

Роль:

Логин:

Пароль:

Добавить

Удалить работника

Паспорт:

Удалить

Таблица работников в депо

Депо	Паспорт	З/п	Профессия
А	40615497	25000	Электрик
С	40973395	35000	Электрик
В	42558262	222233	Водитель

Удалить работу работника

Паспорт:

Удалить

Добавить работу работнику

Депо:

Паспорт:

З/п:

Профессия:

Добавить

Сменить работу работника

Депо:

Паспорт:

З/п:

Профессия:

## Для менеджера-диспетчера

### Городские маршруты

Номер	Начало	Конец	Информация
11	Сосновая ул.	Автовокзал	Продлён до ДК Космос
14	Новая ул.	Первомайский пер.	
24	Тихая ул.	Набережная ул.	
T3	Партизанская ул.	Луговая ул.	
T5	Заслонова ул.	Весенняя ул.	Отменён
T9	Радужная ул.	Чапаева ул.	
31	Северная больница	ЦУМ	
34	Озерный пер.	Песчаная ул.	

### Таблица транспорта в депо

idТехники	idДепо	Номер техники в этом депо
1	A	3
2	A	2
3	B	1
4	C	2

#### Удалить транспорт из депо

idТехники:

Удалить

#### Добавить транспорт в депо

idДепо:

idТехники:

Добавить

### Транспорт на маршруте

Маршрут	idТехники	Рабочий	Тип смены
11	2	41419035	even
T5	4	43807849	morning
11	2	40973395	even
11	3	44311258	morning

#### Удалить со смены

idТехники:

Удалить

#### Добавить транспорт на смену

idПаспорта:

idТехники:

Маршрут:


☐ Утро ☐ Вечер

Добавить

# Страница администратора

На странице работника можно выполнять различные запросы

## Обработка SELECT-запроса



ЗАПРОС

### Введите SQL запрос

SELECT \* FROM employer

Отправить


© 2021 Ренжин Андрей

### Введите SQL запрос

Отправить

12334567	Я	Стачек пр., д. 111	228-228	admin	admin	1234
40151782	Зуйков Яков	Майская ул., д. 19 кв.217	440-656	manager	yy-201	JDsa9S
40615497	Чучумашева Римма	Радужная ул., д. 13 кв.89	450-443	worker	Erellici	oiWoeQ
40973395	Липин Александр	Калинина ул., д. 15 кв.156	560-257	worker	Naria	xiO91V
41372958	Лагутова Марьяна	Школьная ул., д. 15 кв.103	120-157	manager	Huongar	tjsMhF
41419035	Петров Максим	Интернациональная ул., д. 17 кв.60	690-255	worker	Jesi	RerhpR
41467212	Оропай Александра	Южная ул., д. 19 кв.24	630-357	worker	Zanaki	4Mvplg
42558262	Богомазов Афанасий	Колхозный пер., д. 14 кв.56	440-439	worker	Fonsenia	wdjENu
43397762	Карюкина Альбина	Заводская ул., д. 2 кв.109	860-548	worker	Yetti	mTP9rb
43464621	Усоева Анастасия	Победы ул., д. 7 кв.112	480-158	manager	Lari	Q8Xeil
43807849	Поджио Ева	Новая ул., д. 18 кв.57	290-970	worker	Lavado	JPHgxa
43944478	Булатова Любовь	Садовый пер., д. 9 кв.1	950-969	worker	Caro	PV1Gu4
43998660	Овсянников Михаил	Радужная ул., д. 4 кв.40	690-864	worker	Onan	1T0wnA
44311258	Ховрунова Марианна	3 Марта ул., д. 14 кв.122	400-560	worker	ielle	kcQkYm
45163717	Енотин Валентин	17 Сентября ул., д. 1 кв.130	360-672	worker	Shida	69DKx1
45579582	Котов Максим	Трудовая ул., д. 20 кв.81	280-828	worker	Telysani	fLMD4k

## SELECT-запрос с объединением

 ЗАПРОС

**Введите SQL запрос**

```
SELECT e.*, ed.* FROM employee e
INNER JOIN emp_dep ed ON
e.id_passport=ed.id_passport
```


Отправить

**Введите SQL запрос**

Отправить

40615497	Чучумашева Римма	Радужная ул., д. 13 кв.89	450-443	worker	Erellici	olWoeQ	A	40615497	25000	Электрик
40973395	Липин Александр	Калинина ул., д. 15 кв.156	560-257	worker	Naria	xiO91V	C	40973395	35000	Электрик
42558262	Богомазов Афанасий	Колхозный пер., д. 14 кв.56	440-439	worker	Fonsenia	wdjENu	B	42558262	222233	Водитель

## Обработка запроса на Вставку, Изменение, Удаление в таблицу



**Введите SQL запрос**

```
IF EXISTS (SELECT * from employer where id_passport=48566851) DELETE FROM employer
WHERE id_passport=48566851
```

Отправить



### Введите SQL запрос

Изменения применены

### Обработка ошибки

В случае неверного запроса, приложение выводить какая именно ошибка произошла



### Введите SQL запрос

### Введите SQL запрос

Возникло исключение: Недопустимое имя объекта "people".

## Код страницы

```
@{
    Layout = "~/Views/Shared/_LayoutManager.cshtml";
}

<h2>Введите SQL запрос</h2>

<form action="Index" method="post">
    <textarea id="names" name="query"
        type="text" value=""></textarea>
    <button>Отправить</button>
</form>
@if (ViewBag.adm != null)
{
    if (ViewBag.adm.GetType() != typeof(System.String))
    {
        <table border="1" align="center" cellpadding="15">
            @foreach (var d in ViewBag.adm)
            {
                <tr>
                    @for (int i = 0; i < d.Length; i++)
                    {
                        <td>@d[i]</td>
                    }
                </tr>
            }
        </table>
    }
    else
    {
        <p>@ViewBag.adm</p>
    }
}
```

## Пояснения

ViewBag.adm.GetType() != typeof(System.String)

- узнать тип данных, занесённых во ViewBag

```
<form action="Index" method="post">
    <textarea id="names" name="query"
        type="text" value=""></textarea>
    <button>Отправить</button>
</form>
```

- считывает запрос, который формирует администратор

## Код мастер-страницы

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Рабочий профиль</title>
    <link rel="stylesheet" href="/Content/Style.css">
</head>

<body style="overflow-x:hidden;">
```

```

<header class="header">
    <div class="header__selection">
        <div class="logo" style="float: left; margin-top: 6px; margin-left: 215px;">

            <a href="~/Home/Index"></a>
        </div>
    </div>
    @if (Session["status"].ToString() == "manager")
    {
        <nav class="nav" style="margin-right: 215px;">
            <a class="nav__link" href="~/Manager/Profile_">Профиль</a>
            <a class="nav__link" href="~/Manager/WorkPlace">Рабочее место</a>
        </nav>
    }
    else if (Session["status"].ToString() == "worker")
    {
        <nav class="nav" style="margin-right: 215px;">
            <a class="nav__link" href="~/Work/Profile_">Профиль</a>
            <a class="nav__link" href="~/Work/Profession">Профессия</a>
            <a class="nav__link" href="~/Work/Members">Список сотрудников</a>
        </nav>
    }
    else
    {
        <nav class="nav" style="margin-right: 215px;">
            <a class="nav__link" href="~/Admin/Index">Запрос</a>
        </nav>
    }
</header>

<div class="main">
    @RenderBody()
</div>

<footer class="footer">
    © 2021 Ренжин Андрей
</footer>
</body>
</html>

```

## Метод страницы

```

private QueryExc qe = new QueryExc(); // объявление объекта класса QueryExc

// GET: Admin
[HttpPost]
public ActionResult Index(string query) // метод обработки SQL-запроса
{
    ViewBag.adm = qe.SelectorQuery(query); // использование метода
    // вспомогательного класса и передача результата во ViewBag
    return View(); // возвращение в представление
}

```

## Вспомогательный класс

```

public class QueryExc
{
    SqlConnection connection =

```

```

        new SqlConnection("Data Source=localhost;Initial Catalog=autopark;Integrated
Security=True;MultipleActiveResultSets=True;Application Name=EntityFramework"); //
установка строки подключения

    public object SelectorQuery(string queryStr)
    {
        if (queryStr.StartsWith("SELECT"))// определение первого слова запроса
        {
            return Select(new SqlCommand(queryStr, connection)); // объявление
нового запроса на основе строки, и подключения и вызов соответствующего метода по
обработке запроса
        }
        else
        {
            return InsUpdDel(new SqlCommand(queryStr, connection)); // объявление
нового запроса на основе строки и вызов соответствующего метода по обработке запроса
        }
    }

    public object Select(SqlCommand cmd) // метод обработки select-запроса
    {
        using ( connection )
        {
            connection.Open(); // подключение к базе данных
            try
            {
                using (var reader = cmd.ExecuteReader())
                {
                    var model = Read(reader).ToList(); // использование
вспомогательного метода Read

                    connection.Close(); // отключение от базы данных

                    return model; // возвращение списка значений по результатам
запроса
                }
            }
            catch (Exception ex) // действия при возникновении ошибки
            {
                connection.Close();
                return String.Format("Возникло исключение: {0}", ex.Message); //
возвращение содержания ошибки, т.к. тип возвращаемого объекта – object, позволяет нам
вернуть как строку так и список объектов
            }
        }
    }

    private static IEnumerable<object[]> Read(DbDataReader reader) // вспомогательный
метод
    {
        while (reader.Read())
        {
            var values = new List<object>();
            for (int i = 0; i < reader.FieldCount; i++)
            {
                values.Add(reader.GetValue(i));
            }
            yield return values.ToArray(); // вспомогательный yield используется для
создания генераторов последовательностей элементов
        }
    }

    public object InsUpdDel(SqlCommand cmd) // метод обработки запроса на вставку,
удаление, обновление элементов в таблицу
    {

```



```
using (connection)
{
    connection.Open();
    try
    {
        cmd.ExecuteNonQuery(); // выполнение запроса
        connection.Close();
    }
    catch (Exception ex) // действия при возникновении ошибки
    {
        connection.Close();
        return (String.Format("Возникло исключение: {0}", ex.Message));
    }
}
return ("Изменения применены"); // сообщение при успешном внесении изменений
}
```