**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа № 4

дисциплина: Технологии web-программирования

тема: «Разработка и проектирование базы данных»

Выполнил: ст. гр. ПВ-42

Курмачев Сергей Васильевич

Проверил: Картамышев С. В.

Белгород 2020

**Цель:** изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.).

**Задание к работе**

1. Выбрать подходящую СУБД.
2. Изучить методы взаимодействия web-приложения с базой данных (ORM, Active Record).
3. Разработать структуру базы данных.
4. Разработать соответствующие модели в приложении.
5. В отчёт приложить схему базы данных, а также код одной из моделей (на своё усмотрение).

**Ход работы**

Для создания таблиц в базе данных использовались миграции.

*Create\_cars\_table.php:*

<?php  
  
use yii\db\Migration;  
  
*/\*\*  
 \* Handles the creation of table `{{%cars}}`.  
 \*/*class m201222\_143521\_create\_cars\_table extends Migration  
{  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeUp()  
 {  
 $this->createTable('{{%cars}}', [  
 'id' => $this->primaryKey(),  
 'model' => $this->string(128) -> notNull() ->comment('Марка'),  
 'image' => $this->string(128) -> notNull() ->comment('Фотография'),  
 'release\_year' => $this->integer() -> notNull() -> comment('Год выпуска'),  
 'mileage' => $this->integer() ->notNull()->comment('Пробег'),  
 'createdAt' => $this->dateTime()->notNull()->comment('Дата создания'),  
 'updatedAt' => $this->dateTime()->comment('Дата изменения')  
 ]);  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeDown()  
 {  
 $this->dropTable('{{%cars}}');  
 }  
}

*Create\_trips\_types\_table.php:*

<?php  
  
use yii\db\Migration;  
  
*/\*\*  
 \* Handles the creation of table `{{%trips\_types}}`.  
 \*/*class m201224\_115543\_create\_trips\_types\_table extends Migration  
{  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeUp()  
 {  
 $this->createTable('{{%trips\_types}}', [  
 'id' => $this->primaryKey(),  
 'name' => $this->string(128) -> notNull() ->comment('Название'),  
 'max\_price' => $this->integer() -> notNull() -> comment('Максимальная цена'),  
 'description' => $this->text()->notNull()->comment('Описание'),  
 'createdAt' => $this->dateTime()->notNull()->comment('Дата создания'),  
 'updatedAt' => $this->dateTime()->comment('Дата изменения')  
 ]);  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeDown()  
 {  
 $this->dropTable('{{%trips\_types}}');  
 }  
}

*Create\_users\_table.php:*

<?php  
  
use yii\db\Migration;  
  
*/\*\*  
 \* Handles the creation of table `{{%users}}`.  
 \*/*class m210106\_103208\_create\_users\_table extends Migration  
{  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeUp()  
 {  
 $this->createTable('{{%users}}', [  
 'id' => $this->primaryKey(),  
 'login' => $this->string(512)->notNull()->comment('Логин'),  
 'password' => $this->string(512)->notNull()->comment('Пароль'),  
 'accessToken' => $this->string(512)->comment('Токен пользователя')  
 ]);  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeDown()  
 {  
 $this->dropTable('{{%users}}');  
 }  
}

*Create\_reviews\_table.php:*

<?php  
  
use yii\db\Migration;  
  
*/\*\*  
 \* Handles the creation of table `{{%reviews}}`.  
 \*/*class m210113\_105327\_create\_reviews\_table extends Migration  
{  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeUp()  
 {  
 $this->createTable('{{%reviews}}', [  
 'id' => $this->primaryKey(),  
 'author' => $this->string(128)->notNull()->comment('Название'),  
 'author\_status' => $this->string(128)->notNull()->comment('Статус относительно службы такси'),  
 'picture' => $this->string(128) -> notNull() ->comment('Картинка'),  
 'text' => $this->string(256) -> notNull() ->comment('Текст отзыва'),  
 'createdAt' => $this->dateTime()->notNull()->comment('Дата создания'),  
 'updatedAt' => $this->dateTime()->comment('Дата изменения')  
 ]);  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function safeDown()  
 {  
 $this->dropTable('{{%reviews}}');  
 }  
}

В результате база данных стала выглядеть следующим образом:









С помощью генератора Yii/gii были сгенерированы модели для таблиц базы данных. Например, модель Car:

<?php  
  
namespace app\modules\v1\models;  
  
use Yii;  
use yii\behaviors\SluggableBehavior;  
  
*/\*\*  
 \* This is the model class for table "cars".  
 \*  
 \** ***@property*** *int $id  
 \** ***@property*** *string $model Марка  
 \** ***@property*** *string $image Фотография  
 \** ***@property*** *int $release\_year Год выпуска  
 \** ***@property*** *int $mileage Пробег  
 \** ***@property*** *string $createdAt Дата создания  
 \** ***@property*** *string|null $updatedAt Дата изменения  
 \** ***@property*** *string $url URL  
 \*/*class Car extends BaseModel  
{  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public static function tableName()  
 {  
 return 'cars';  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function rules()  
 {  
 return [  
 [['model', 'image', 'release\_year', 'mileage'], 'required'],  
 [['release\_year', 'mileage'], 'integer'],  
 [['createdAt', 'updatedAt'], 'safe'],  
 [['model', 'image', 'url'], 'string', 'max' => 128],  
 ];  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function attributeLabels()  
 {  
 return [  
 'id' => 'ID',  
 'model' => 'Марка',  
 'image' => 'Фотография',  
 'release\_year' => 'Год выпуска',  
 'mileage' => 'Пробег',  
 'createdAt' => 'Дата создания',  
 'updatedAt' => 'Дата изменения',  
 'url' => 'URL',  
 ];  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function behaviors()  
 {  
 $behaviors = parent::*behaviors*();  
  
 $behaviors[] = [  
 'class' => SluggableBehavior::class,  
 'attribute' => 'model',  
 'slugAttribute' => 'url'  
 ];  
  
 return $behaviors;  
 }  
  
 */\*\*  
 \* {****@inheritdoc****}  
 \*/* public function toArray(array $fields = [], array $expand = [], $recursive = true)  
 {  
 return [  
 'id' => $this->id,  
 'model' => $this->model,  
 'image' => $this->image,  
 'release\_year' => $this->release\_year,  
 'mileage' => $this->mileage,  
 'createdAt' => $this->createdAt,  
 'url' => $this->url  
 ];  
 }  
}

**Вывод:** в результате выполнения данной работы были изучены основы взаимодействия web-приложения с базой данных, спроектирована база данных для хранения информации приложения (страницы, пользователи и т.п.).