

Одесский национальный политехнический университет

Институт компьютерных систем

Кафедра информационных систем

Лабораторная работа №11

По дисциплине: «Разработка сайта с использованием MVC фреймворка»

Тема: «Основы языка PHP»

Выполнил:

Студент группы АИ-172у

Терентьев К. А.

Проверили:

Червоненко П.П.

Панькина А.С.

Одесса 2019

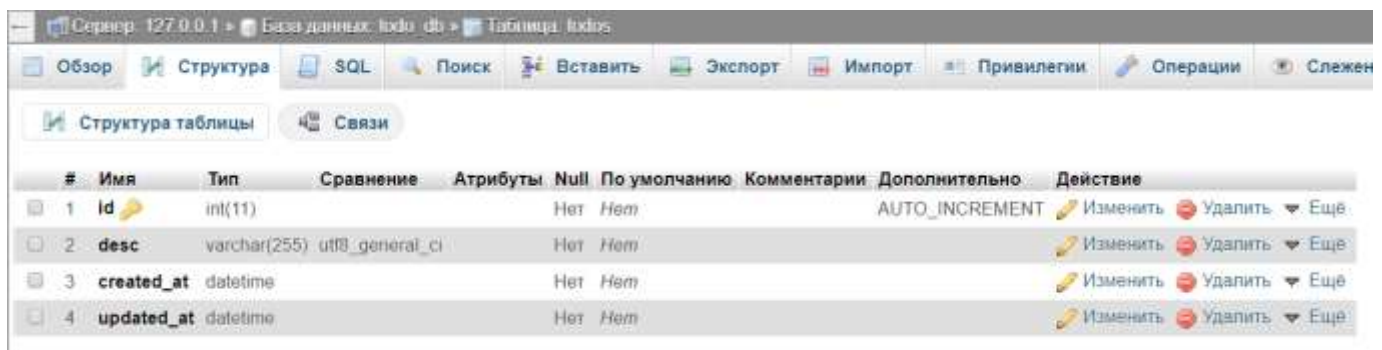
Тема: Разработка сайта с использованием MVC фреймворка.

Цель: получить практические навыки разработки модулей приложения с помощью фреймворка CodeIgniter.

Ход работы:

1. Загрузим архив с базовым проектом CodeIgniter с сайта <https://www.codeigniter.com/>
2. Следуя паттерну MVC, создадим небольшое приложение для заметок – TODO List.

Сначала создадим базу данных для приложения. В ней будут храниться данные о заметках.



The screenshot shows a database management interface with a table structure view. The table is named 'todos' and has four columns: 'id', 'desc', 'created_at', and 'updated_at'. The 'id' column is an integer (11) with an auto-increment attribute. The 'desc' column is a varchar(255) with a utf8_general_ci collation. The 'created_at' and 'updated_at' columns are datetime types. The interface includes tabs for 'Обзор', 'Структура', 'SQL', 'Поиск', 'Вставить', 'Экспорт', 'Импорт', 'Привилегии', 'Операции', and 'Слежен'. Below the table structure, there are buttons for 'Структура таблицы' and 'Связи'.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
2	desc	varchar(255) utf8_general_ci			Нет	Нет			Изменить Удалить Ещё
3	created_at	datetime			Нет	Нет			Изменить Удалить Ещё
4	updated_at	datetime			Нет	Нет			Изменить Удалить Ещё

Создадим в директории model класс модели Todo_model.php. Он будет наследоваться от класса MY_Model.php (https://github.com/avenirer/CodeIgniter-MY_Model), который в свою очередь, является общедоступным классом, наследующимся от CI_Model.php (и содержит множество методов для работы с базой данных. Так, операции CRUD в нем уже реализованы).

Листинг 1 – класс Todo_model.php

```
<?php class Todo_model extends MY_Model
{
    public function __construct()
    {
        parent::__construct();
        $this->return_as = "object";
    }
}
```

Теперь создадим класс контроллера, который будет реагировать на действия пользователя. Класс контроллера также наследуется от MY_Controller.

Листинг 2 – класс-контроллер Todolist.php

```
<?php
```

```
class Todolist extends MY_Controller
{

    public function __construct()
    {
        parent::__construct();
        $this->load->model('todo_model');
    }

    public function index()
    {
        // get data from the model
        $items = $this->todo_model->get_all();
        $this->data['items'] = $items;
        // pass the data to the view
        $this->load->view('todolist', $this->data);

    }

    public function todo($id)
    {
        $data["item"] = $this->todo_model->get($id);
        $this->load->view('todo', $data);
    }

    public function add_task()
    {
        // get values from the post
        $item = $this->input->post('item');

        $insert = array();
        $insert['desc'] = $item;
        // insert them into the database
        $id = $this->todo_model->insert($insert);

        if ($id) {
            $items = $this->todo_model->get_all();

            $this->data['items'] = $items;
            $this->load->view('todolist', $this->data);
        } else {
            echo "did not record values";
        }
    }
}
```

```

    }
}

function edit_task($id)
{
    if ($id) {
        $item = $this->input->post('item');
        $update = array();
        $update['id'] = $id;
        $update['desc'] = $item;
        $this->todo_model->update($update, $id);
    }
    $this->index();
}

function delete_task($id)
{
    $this->todo_model->delete($id);
    $this->index();
}
}

```

И наконец, перейдем к созданию представлений, или View. Сайт будет состоять из двух страничек, одна – со списком заданий, на ней можно будет добавить новое задание, удалить одно из существующих и открыть детали конкретного задания, и вторая – с деталями каждого задания, где текст задания можно будет изменить и сохранить, вернуться назад к списку и удалить данное задание.

Листинг 3 – view todolist.php

```

<!DOCTYPE HTML>
<meta charset="UTF-8">
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>TODO</title>
    <link rel="stylesheet" href="<?php echo site_url('assets/bootstrap-3.3.7-
dist/css/bootstrap.min.css'); ?>">
    <link rel="stylesheet" href="<?php echo site_url('assets/bootstrap-3.3.7-
dist/css/bootstrap-theme.min.css'); ?>">

```



```

</ul>
</div>
</div>
</div>

<div class="col-md-6" id="add_form">
<div class="panel">
<div class="panel-heading">
    <h3>Добавить новое задание</h3>
</div>
<div class="panel-body">
    <form method="POST" action="<?php echo site_url('todolist/add_task');
?>">
        <input type="text" placeholder="Задание" name="item" required>
        <br/><br/>
        <button class="btn btn-primary" type="submit">Добавить</button>
    </form>
</div>
</div>
</div>
</div>

</body>
</html>

```

Листинг 5 – view todo.php

```

<!DOCTYPE HTML>
<meta charset="UTF-8">
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>TODO</title>
    <link rel="stylesheet" href="<?php echo site_url('assets/bootstrap-3.3.7-
dist/css/bootstrap.min.css'); ?>">
    <link rel="stylesheet" href="<?php echo site_url('assets/bootstrap-3.3.7-
dist/css/bootstrap-theme.min.css'); ?>">
    <link rel="stylesheet" href="<?php echo site_url('assets/font-awesome-
4.7.0/css/font-awesome.min.css'); ?>">
    <link rel="stylesheet" href="<?php echo site_url('assets/animate-
css/animate.min.css'); ?>">
    <link rel="stylesheet" href="<?php echo site_url('assets/hover-

```

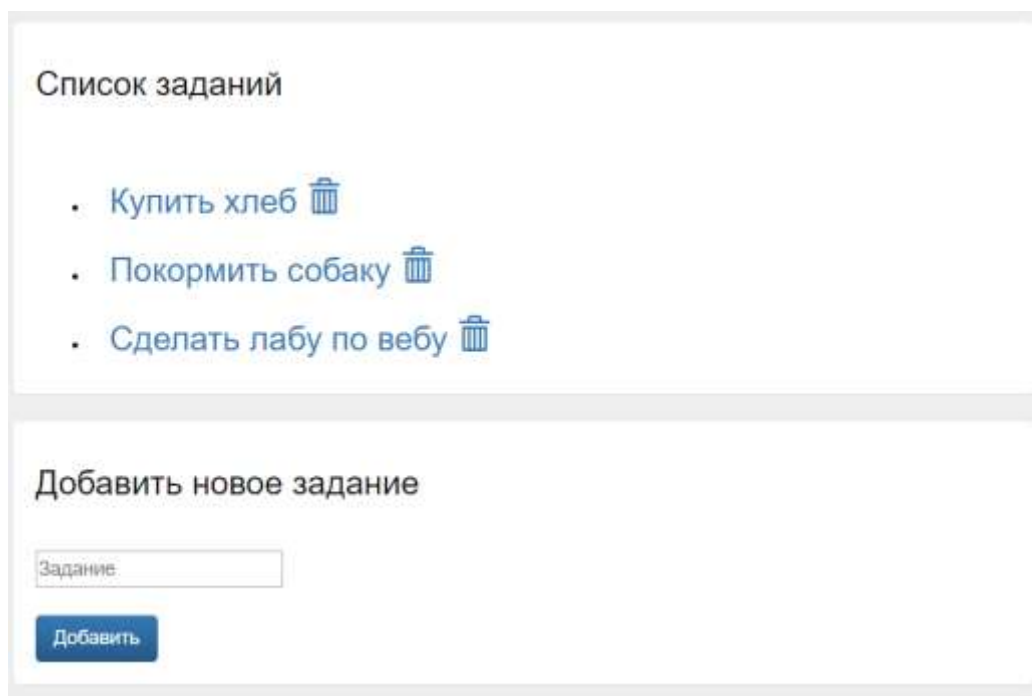
```

css/hover.min.css'); ?>">
    <link rel="stylesheet" href="<?php echo site_url('assets/css/style.css');
?>">
</head>
<body>
<div class="container" id="main-container">
<div class="row">
<div class="col-xs-12">
<div class="jumbotron">
    <h1 class="textcenter">Todo</h1>
</div>
</div>
</div>
<div class="row">
<div class="col-md-12" id="edit_form">
<div class="panel">
<div class="panel-body">
    <form method="POST"
        action="<?php echo site_url('todolist/edit_task/' . $item->id);
?>">
        <input type="text" name="item" required value="<?php echo $item-
>desc; ?>">
        <br>
        <?php echo 'Created at: ' . $item->created_at; ?>
        <?php if ($item->updated_at != '0000-00-00 00:00:00') {?><br><?php
echo 'Updated at: ' . $item->updated_at; } ?>
        <br/><br/>
        <button class="btn btn-secondary" id="back_but" type="button"
            onclick="location.href='http://localhost/todo-
list/todolist'">Назад
        </button>
        <button class="btn btn-primary" id="edit_but"
type="submit">Сохранить</button>
    </form>
    <br>
    <form method="POST" action="<?php echo site_url('todolist/delete_task/' .
$item->id); ?>">
        <button class="btn btn-danger" id="delete_but"
type="submit">Удалить</button>
    </form>
</div>
</div>
</div>
</div>
</div>




```

```
</body>
</html>
```

Результат:

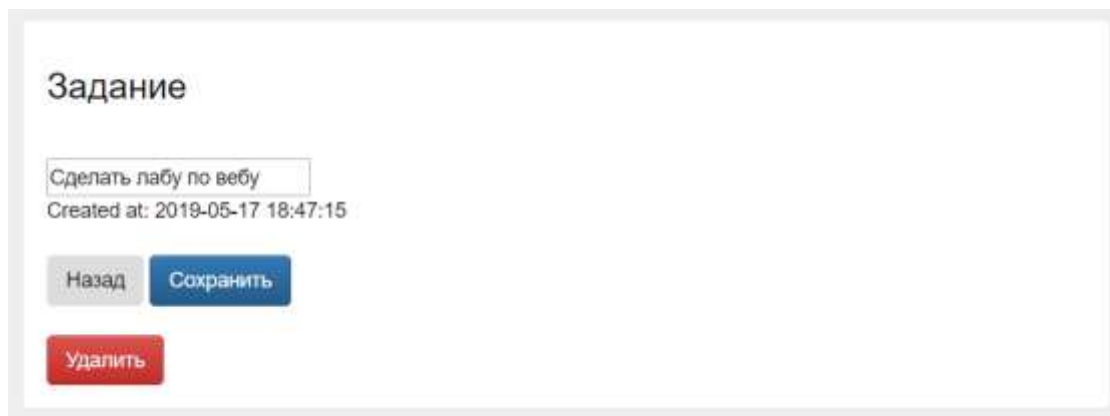


Список заданий

- Купить хлеб 
- Покормить собаку 
- Сделать лабу по вебу 

Добавить новое задание

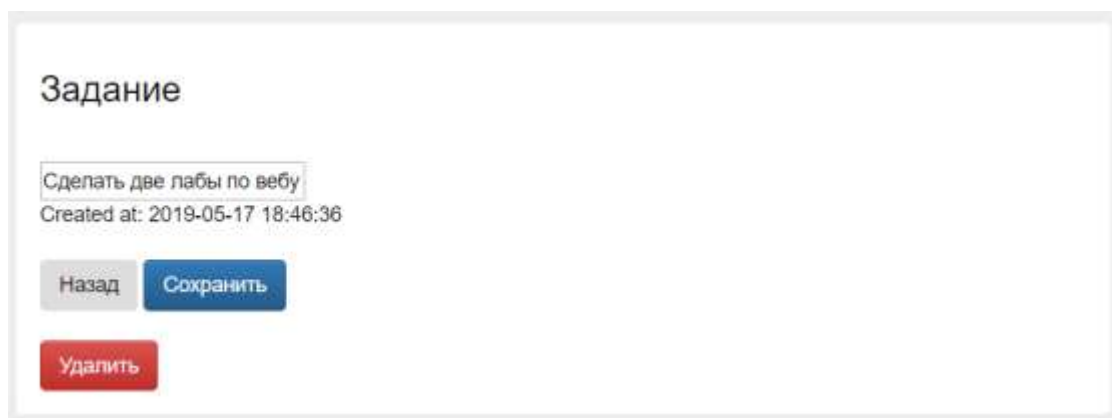
Рисунок 1 – Список задач



Задание

Created at: 2019-05-17 18:47:15

Рисунок 2 – Детали задачи



Задание

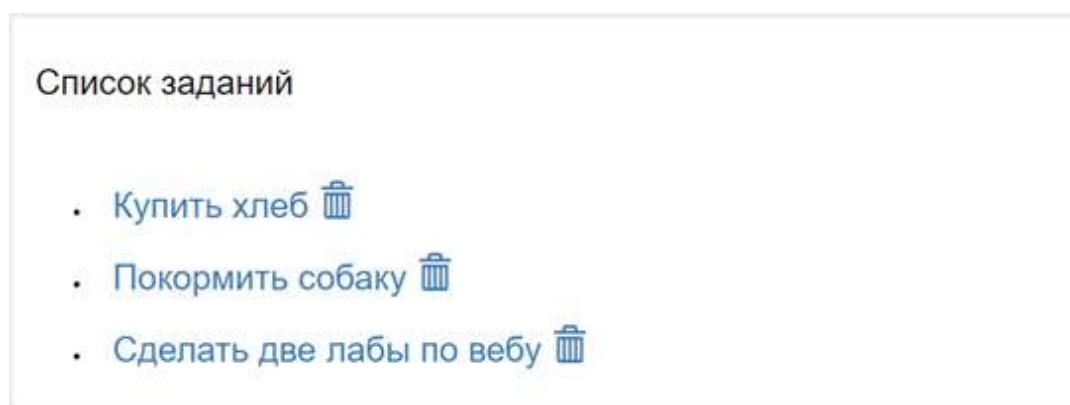
Сделать две лабы по вебу

Created at: 2019-05-17 18:46:36

Назад Сохранить

Удалить

Рисунок 3 – Изменение задачи



Список заданий




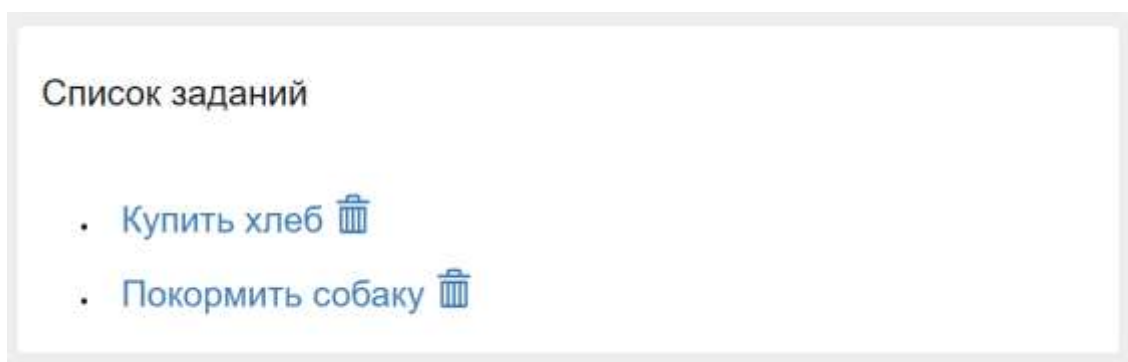
- Купить хлеб 
- Покормить собаку 
- Сделать две лабы по вебу 

Рисунок 4 – Изменения в списке



Список заданий


- Купить хлеб 
- Покормить собаку 

Рисунок 5 – Результат удаления записи

Выводы:

В ходе выполнения лабораторной работы мы научились пользоваться фреймворком CodeIgniter и создавать на нем сайты с помощью паттерна MVC. Также в работе использовались библиотеки Bootstrap.