МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЁТ**

о выполнении лабораторной работы № 8

по дисциплине «WEB-технологии»

Выполнил: ст. гр. ИС/б-19-2-о

Газукин А. С.

Проверил:

**Куркчи А. Э.**

Севастополь, 2021 г.

**ЛАБОРАТОРНАЯ РАБОТА № 8 «ИССЛЕДОВАНИЕ АРХИТЕКТУРЫ MVC ПРИЛОЖЕНИЯ И ВОЗМОЖНОСТЕЙ ОБРАБОТКИ ДАННЫХ HTML-ФОРМ НА СТОРОНЕ СЕРВЕРА С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА PHP»**

**1 ЦЕЛЬ РАБОТЫ**

Исследовать особенности построения приложений с использованием архитектуры MVC. Изучить основы синтаксиса PHP и принципы функционирования MVC приложения на стороне сервера. Приобрести практические навыки использования языка PHP для генерации HTML-кода и обработки HTML-форм в MVC приложении.

**2 ПОСТАНОВКА ЗАДАЧ**

1. В соответствии с рекомендациями, представленными в п. 1.2.8 реструктурировать код персонального сайта на основе шаблона MVC.

Внимание! Запрещается использовать готовые решения для реализации MVC архитектуры.

2. Для страницы «Фотоальбом» реализовать хранение данных фото (подписей и имен файлов) в соответствующих константах модели Photo. Вывод таблицы, содержащей фото, реализовать в представлении с использованием операторов циклов.

3. Для страницы «Мои интересы» реализовать хранение данных (название категорий, интересов, описание интересов) в соответствующих константах модели Interest. Вывод меню ссылок и интересов реализовать в представлении с использованием операторов циклов.

4. Реализовать класс FormValidation, выполняющий валидацию данных форм, передаваемых на сторону сервера. Рекомендуемая структура класса:

- Rules – поле(массив), содержащее набор правил для проверки валидности данных;

- Errors – поле(массив), содержащее тексты ошибок возникших при проверке валидности данных;

- isNotEmpty(data) – метод проверки является ли значение data не пустым – возвращает сообщение об ошибке, если таковая имеется;

- isInteger(data) – метод проверки является ли значение data строковым представлением целого числа – возвращает сообщение об ошибке, если таковая имеется;

- isLess(data, value) – метод проверки является ли значение data строковым представлением целого числа и не меньшим, чем value – возвращает сообщение об ошибке, если таковая имеется;

- isGreater(data, value) – метод проверки является ли значение data строковым представлением целого числа и не большим, чем value – возвращает сообщение об ошибке, если таковая имеется;

- isEmail(data) – метод проверки является ли значение data строковым представлением email – возвращает сообщение об ошибке, если таковая имеется;

- SetRule(field\_name, validator\_name) – метод, добавляющий в массив Rules проверку для поля field\_name типа validator\_name;

- Validate(post\_array) – метод выполняющий проверку элементов в массиве post\_array, в соответствии с правилами Rules и сохраняющий сообщения об ощибках в поле Errors;

- ShowErrors() – метод, выводящий все сообщения об ошибках из поля Errors в формате HTML.

5. С использованием разработанного класса реализовать валидацию форм «Контакт» и «Тест по дисциплине “…”».

6. Реализовать дочерний класс CustomFormValidation от класса FormValidation, дополнив его возможностью выполнения специализированной проверки формы «Тест по дисциплине» на стороне сервера.

7. Реализовать дочерний класс ResultsVerification от класса CustomFormValidation, дополнив его возможностью проверки правильности ответов, введенных пользователем на странице "Тест по дисциплине" (реализовать проверку правильности для вопросов с элементами ввода типа RadioButton, ComboBox или однострочный текст) и вывода результатов проверки пользователю.

**3 ХОД РАБОТЫ**

1) Весь код проекта реструктурирован под MVC архитектуру.

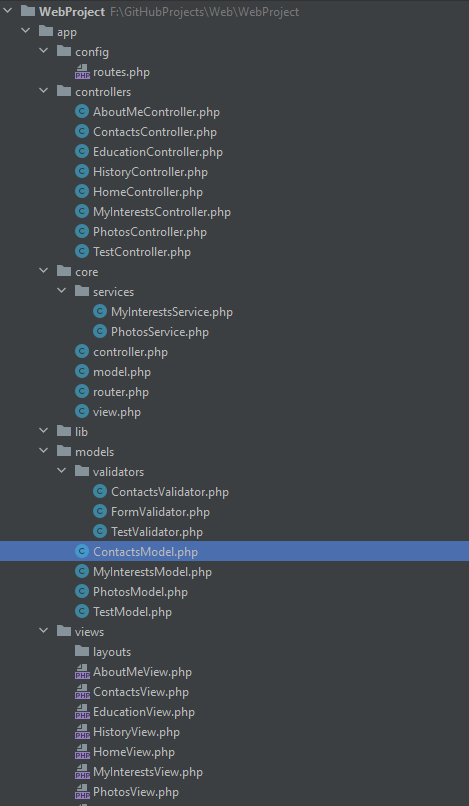


Рисунок 1 – Архитектура MVC

2) Список картинок из галереи был перенесен на язык РHP.

3) Список интересов был перенесен на язык PHP.

4) Был написан класс FormValidator, описывающий реализацию методов: isNotEmpty(); isInteger(); isMinAnswSixe(); isLess(); isGreater(); isEmail(); isPhone(); isWord(); isDate(); setRule(); validateRule(); validate(); showErrors();

5) Также были описаны классы ContactsValidator, TestValidator предопределяющие эти методы.

6) Был написан функционал на стороне сервера, проверяющий правильность введённых ответов на странице «Тест».

**4 ВЫВОДЫ**

В ходе выполнения лабораторной работы были исследованы особенности построения приложений с использованием архитектуры MVC. Изучены основы синтаксиса PHP и принципы функционирования MVC приложения на стороне сервера. Приобретены практические навыки использования языка PHP для генерации HTML-кода и обработки HTML-форм в MVC приложении. Код представлен в репозитории Github: https://github.com/Xelager/Web