МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГАОУ ВО «СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

ОТЧЕТ

о выполнении лабораторной работы № 1

«Исследование архитектуры MVC приложения и возможностей обработки данных HTML-форм на стороне сервера с использованием языка PHP.»

по дисциплине

«Веб-технологии»

Выполнил: ст.гр. ПИ/б-22-2-о Базарный А.Р.

Проверил:

Старший преподаватель ИС

Дрозин А. Ю.

Севастополь, 2025

**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.

**Постановка задач:**

1. В соответствии с рекомендациями, представленными в п. 1.2.8 реструктурировать код пресонального 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; 47 - 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.

**Ход работы**

Для работы с PHP был использован OSP Open Server Panel, иерархия папок показана на рисунках 1, 2.



Рисунок 1 – Иерархия папок



Рисунок 2 – Иерархия папок

После создания шаблона для работы с MVC шаблоном для каждой страницы из 4 лабораторной были созданы модель, представление и контроллер.

В модели “Фото” были использованы константные переменные для вывода фотографий на экран с помощью контроллера, это показано на рисунке 3.



Рисунок 3 – Модель фото

Для страницы “Мои интересы” также была реализована модель, хранящая основные данные. Весь функционал, написанный ранее в js был переписан с использованием PHP, открытие больших фото осталось в js.



Рисунок 4 – Мои интересы

Для валидации в классе FormValidation был реализован весь функционал, что показано на рисунке 5:

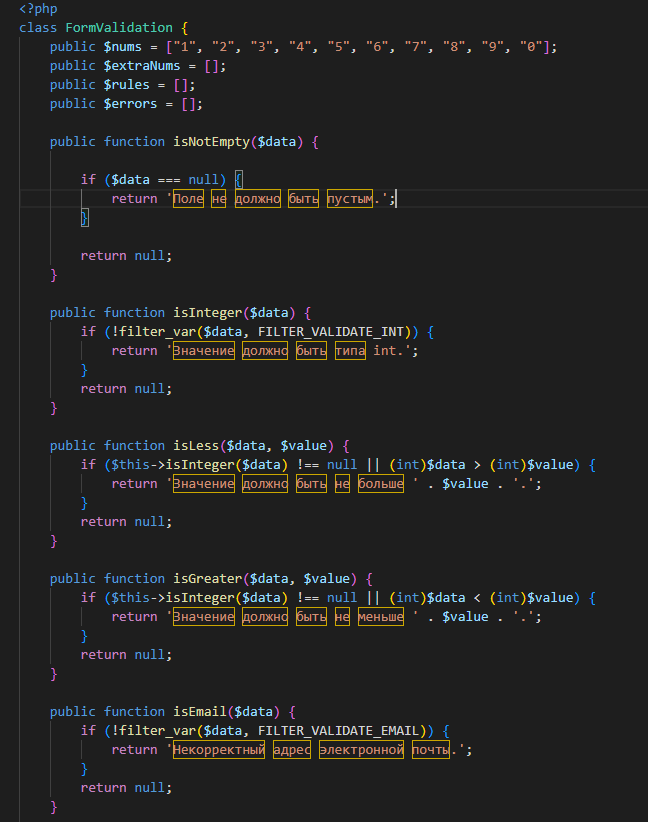


Рисунок 5 – Валидация

С помощью этого класса было переработано представление страниц контакты и тест по дисциплине. Для страницы “Тест по дисциплине” были реализованы 2 дополнительных валидатора для проверки вводимых данных и для проверки результатов.

**Вывод:** В ходе лабораторной работы был переписан сайт, созданный ранее на язык PHP с использованием MVC модели, весь функционал был воссоздан, Большинство js скриптов заменено.