

УДК 004.4'242
DOI: 10.15827/0236-235X.141.175-182

Дата подачи статьи: 25.08.22, после доработки: 09.11.22
2023. Т. 36. № 1. С. 175–182

Модуль генерации панели управления для веб-проектов

С.В. Жуков¹, аспирант, *coder.stas@gmail.com*

А.А. Суслин¹, студент, *andriw_suslin@mail.ru*

О.А. Ковалева¹, д.т.н., доцент, профессор кафедры математического моделирования и информационных технологий, *solomina-oa@yandex.ru*

¹ Тамбовский государственный университет им. Г.Р. Державина,
г. Тамбов, 392036, Россия

В статье рассмотрены проблемы создания панели управления для веб-проектов. Приведена реализация программного модуля, написанного на языке программирования PHP, позволяющего создавать панель администратора для веб-ресурсов без необходимости верстки элементов управления, за счет предустановленного в модуле шаблона Gentelella Admin (разработан и поддерживается Colorlib и AigarsSilkalns). Описанный модуль может использоваться как в проектах, построенных на базе популярной MVC-модели, так и в ресурсах, написанных на базе PHP-фреймворка. Панель управления включает множество типовых элементов управления (поля ввода, переключатели, таблицы, списки и т.п.) и может поддерживать пользовательские компоненты, добавляющие новые элементы управления.

Цель исследования – оптимизация процесса разработки веб-проектов посредством применения созданного модуля. Предметом рассмотрения являются способы реализации административной панели веб-приложения. Представленный модуль позволяет генерировать коды HTML, CSS и JS для визуализации панели управления. Интерфейс пользователя задается в виде PHP-кода. В нем создаются объекты – элементы интерфейса, которые группируются в нужные разработчику формы и панели. Использование модуля позволяет уменьшить время на формирование панели управления, а также создавать приложения без навыков в верстке и организации работы клиентской стороны пользовательского интерфейса.

Теоретическая значимость работы заключается в анализе типовых элементов панели администратора и описании принципов работы созданного модуля по генерации административной панели.

Практическая значимость состоит в уменьшении времени разработки веб-проектов за счет применения модуля генерации панели управления.

Ключевые слова: веб-проект, административная панель, панель управления веб-проектом, генерация интерфейса, элементы управления, MVC-модель, CMS, фреймворк.

Аудитория сети Интернет с каждым годом неуклонно расширяется. Например, в России, по данным Фонда «Общественное мнение», число респондентов, часто использующих Интернет, выросло с 62 % (по данным опроса за 2018 г.) до 74 % (по данным опроса за 2022 г.), а в мире, по данным отчета о состоянии цифровой сферы Global Digital 2022, число пользователей сети увеличилось за последний год на 192 млн человек. На этом фоне все более востребованными становятся и разнообразные ресурсы, представленные в Интернете, растет количество разработчиков ПО. Так, крупнейший хостинг для IT-проектов GitHub за 2021 г. получил 16 млн новых пользователей и 61 млн новых репозитив (<https://octoverse.github.com/>).

Для многих веб-проектов требуется система управления. Одним из вариантов ее реализации может быть так называемая панель администратора (раздел сайта, который предоставляет полный набор возможностей для управления и

настройки веб-ресурса [1]). В статье [2] проведен обзор работ, посвященных внедрению административной панели в веб-проект. Также подробно рассмотрен процесс создания страниц панели администратора с использованием фреймворка Laravel.

Для создания панели управления для одной сущности (страницы сайта) автору потребовалось реализовать три шаблона, два из которых почти идентичны. Для работы веб-сайта могут потребоваться десятки сущностей (товары, сотрудники, новости и т.п.) со сложными структурами (результаты форм обратной связи, торговые предложения и т.п.), но для базового управления необходимы все те же шаблоны списка элементов и детальной страницы элемента. Сами шаблоны, в свою очередь, состоят из повторяющихся элементов (кнопки, поля ввода, поля выбора варианта и т.д.). Таким образом, написание новой панели для нового проекта – задача ресурсоемкая и однообразная.

Для разных веб-ресурсов активно создаются панели управления, о чем можно судить по регистрации патентов на подобные продукты, например [3–5].

Для разработки панели администратора можно применять специальные библиотеки. Обычно они имеют привязку к определенному фреймворку, но позволяют значительно сократить время на разработку программного продукта за счет того, что дают возможность автоматически генерировать интерфейс для пользователя. Подобные библиотеки разрабатываются не только для веб-приложений, но и для мобильных и десктопных решений. Более подробно этот вопрос изложен в [6]. Пример использования библиотеки с привязкой к фреймворку рассмотрен в [7].

Однако при необходимости поддержки легаси-кода, а также при создании прототипа, не работающего на базе одного из современных фреймворков, для разработчика становится актуальной задача генерации панели администратора. На примере работы [8] можно увидеть, как много внимания и времени разработчик уделил адаптации шаблона для мобильных устройств при использовании библиотеки, генерирующей адаптивный интерфейс.

Генерация панелей управления для фреймворков

Задача создания панели управления для веб-проектов часто ставится и перед разработчиками, использующими в работе фреймворки. Фреймворки предоставляют типовую структуру приложения и обычно имеют функционал, позволяющий расширять возможности разрабатываемого продукта за счет подключения отдельных модулей. Поэтому для них написаны разные инструменты, которые позволяют создавать панель администратора. На примере фреймворка Laravel были проанализированы несколько реализаций админ-панелей с открытым исходным кодом и коммерческие продукты.

Nova – коммерческий продукт от разработчиков Laravel, написанный с использованием Vue, дает небольшой инструментарий для кастомизации админ-панели, но требует меньше времени для развертывания [9].

Orchid – инструмент под лицензией MIT, позволяет очень гибко настраивать админ-панель, но требует ручной настройки даже базовых страниц для редактирования элементов (эти проблемы отчасти решаются отдельными пакетами вроде Crud) [10].

Voyager – продукт с открытым исходным кодом, отличается возможностью частичной настройки панели администратора в визуальном режиме, позволяющей реализовать базовые действия (просмотр списка сущностей и конкретной сущности, а также ее редактирование, добавление и удаление) над записями в БД [11].

Сравнение пакетов представлено в таблице.

Пакеты для создания административного интерфейса

A comparison of packages to create an administrative interface

Возможность пакета	Nova	Orchid	Voyager
Добавление своих элементов в административный интерфейс	Да	Да	Да
Создание пользовательской формы (без сторонних решений)	Да	Да	Нет
Использование отдельно от фреймворка	Нет	Нет	Нет
Подключение пользовательских стилей и скриптов	Да	Да	Да

Можно заметить общий недостаток – инструменты разрабатываются под кодовую базу конкретного фреймворка (в данном случае Laravel) и не работают без него. Базовая архитектура фреймворка позволяет создавать такие пакеты и реализовывать в них дополнительные функции, основанные на специфичном для фреймворка коде. При этом существуют общие принципы организации кода, которые также позволяют создать пакет с определенным местом в системе, но без ее привязки к конкретной реализации. Одной из них является распространенная в веб-проектах модель MVC. Создание для нее инструмента с функционалом, схожим с рассмотренными выше пакетами, остается актуальной задачей.

Задачи визуализации элементов панели управления

Были рассмотрены панели администратора в популярных системах управления контентом и определены задачи, наиболее часто решаемые с их помощью для управления веб-ресурсами. По данным компании iTrack, лидерами по количеству использований в Рунете с 2014 по 2021 гг. остаются системы WordPress, 1С-Битрикс, Joomla (<https://itrack.ru/research/>

cmsrate/dynamic/11/60/). По данным ресурса w3techs.com, на июнь 2022 г. в мире лидирующее место среди CMS занимает WordPress, за ним идут Joomla, Drupal (https://w3techs.com/technologies/overview/content_management).

Анализ панелей систем WordPress, Drupal, 1С-Битрикс, Joomla показал, что панель управления состоит из комбинации простых повторяющихся элементов (например, кнопки, поля ввода) и специфичных виджетов, необходимых для конкретных проектов (например, окно просмотра 3D-моделей от сервиса sketchfab.com). Первые будут применяться для решения большинства задач, вторые требуются далеко не всегда. Поэтому модуль для генерации панели администратора должен иметь в своем базовом наборе как можно больше типовых элементов, а также предоставлять возможность добавления виджетов, требующихся конкретному проекту. Для определения списка типовых элементов для разрабатываемого модуля целесообразно обратиться к популярному фреймворку Bootstrap, в документации которого уже имеется описание наиболее часто используемых элементов [12].

Любой веб-ресурс можно представить в виде системы, состоящей из некоторого набора взаимосвязанных элементов. Для отображения набора однотипных элементов в рассматриваемых CMS используются списки, а для отображения иерархических структур – деревья. Так как это наиболее часто используемые виды отображения элементов, необходимо, чтобы модуль имел механизмы по формированию соответствующих представлений.

Кроме элементов интерфейса, пользовательских виджетов, представления наборов элементов, для составления страницы требуется каркас, в котором будут размещаться эти элементы. С его помощью можно осуществлять навигацию между различными группами элементов. Для него подойдут универсальные меню, куда можно добавлять свои пункты, типовые макеты страниц для списков и форм.

Также стандартной задачей при создании административной панели является ограничение доступа к функционалу системы для случайных пользователей. Для ее решения используются три типовые страницы – для авторизации, регистрации и восстановления доступа к учетной записи.

Последняя группа типовых страниц, встречающихся в рассмотренных системах, состоит из шаблонов для отображения ошибок в процессе выполнения кода, а также страниц для

отображения кодов состояния HTTP (например, для отображения одной из наиболее частых ошибок запроса 404).

Приведенные составляющие присутствуют не только в популярных системах управления контентом, но и почти в каждом шаблоне административной панели. Например, платформа themeforest.net на июнь 2022 г. предоставляла выбор из более чем 1 200 шаблонов, 800 из которых используют Bootstrap [13]. Также эти составляющие можно найти в пакетах административных панелей для PHP-фреймворков, например Voyager для Laravel.

На основании типовых задач, решаемых панелью управления, был разработан модуль генерации административной панели на языке PHP, которая позволяет создавать административный интерфейс для любого приложения, построенного на базе MVC-модели или фреймворка на языке PHP.

Структура модуля генерации web-панели администратора

Принцип работы MVC-модели предполагает, что контроллер, основываясь на действиях пользователей и данных из модели, отправляет запрос на формирование определенного представления (рис. 1) [14].

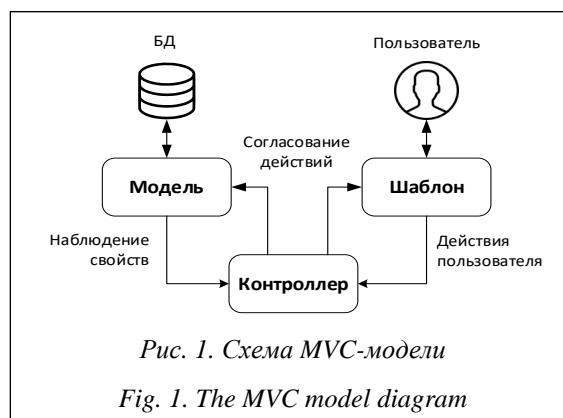


Рис. 1. Схема MVC-модели

Fig. 1. The MVC model diagram

Модуль генерации административного интерфейса работает на уровне представления. Его задача заключается в том, чтобы предоставить пользователю сформированную страницу администратора системы. Принцип работы модуля сводится к следующей схеме.

1. Контроллер получил из модели данные, отобрал нужные и передал их в шаблон формы.

2. Шаблон формы распределил данные от контроллера по элементам интерфейса, вызывая каждый из них для генерации кода страницы.

3. Каждый вызванный элемент интерфейса сформировал свою часть кода и передал ее в шаблон.

4. Шаблон собрал страницу (или ее часть) и передал ее в браузер пользователю.

5. Пользователь проанализировал данные и что-то изменил в них.

6. Информация о действиях пользователя передана в контроллер.

На последнем шаге круг действий замыкается и начинается с начала. В некоторых случаях действия пользователя не отправляются на контроллер, а обрабатываются на локальной машине (если действие не предполагает завершения работы с данными). Описанная схема представлена на рисунке 2.

ницы – набор взаимосвязанных вызовов элементов страницы (A1). Каждый вызов компонента в каркасе запускает процесс формирования html-кода для отображения элемента (A3). Для каждого подготовленного элемента интерфейса подключаются стили и скрипты (A2). Полностью сформированный элемент возвращается в представление и занимает свое место согласно каркасу (A4). На этом этапе происходят объединение стилей и скриптов, а также удаление дублирующихся ресурсов.

Каждый элемент интерфейса может являться составным, то есть во время своего формирования вызывать дополнительные элементы. Имеется возможность предварительно создать общий каркас страницы, а затем расширять его

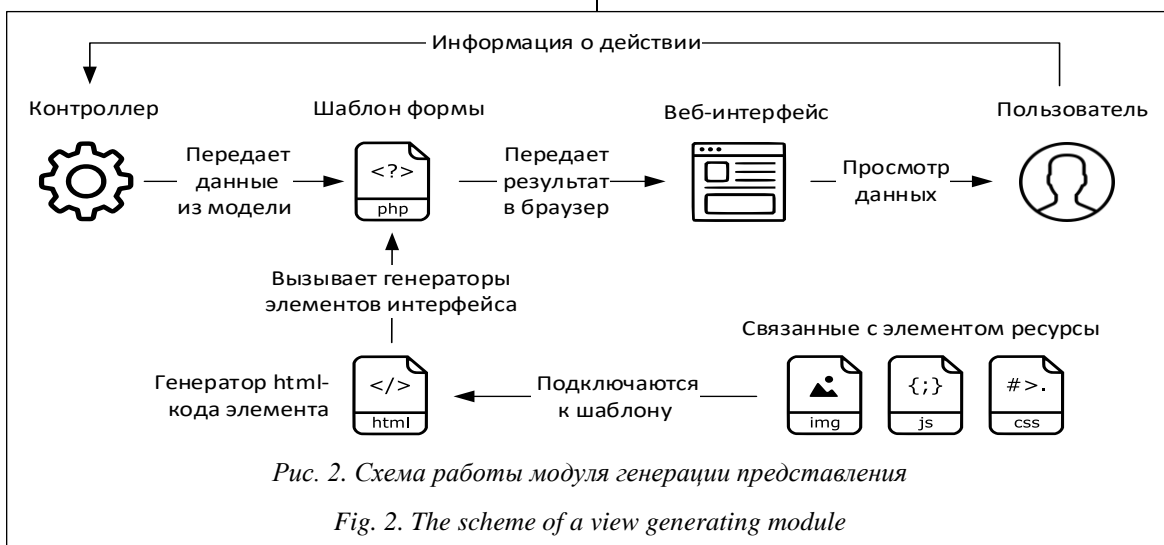


Рис. 2. Схема работы модуля генерации представления

Fig. 2. The scheme of a view generating module

С технической точки зрения данная система реализована посредством группы взаимосвязанных классов, находящихся в едином пространстве имен. Присутствует класс, отвечающий за вызов представления. Класс каждого представления наследуется от его базового класса и содержит описание интерфейса пользователя, заданное в виде сгруппированных элементов интерфейса. Наборы классов, описывающие элементы интерфейса, унаследованы от общего родителя, обеспечивающего работу каскадного вызова стандартных методов элемента интерфейса (стиль кода [15]).

Рассмотрим IDEFO-модель представления, сделанного при помощи модуля генерации админ-панели, и разберем выполняемые им функции (рис. 3).

Представление вызывается контроллером. Согласно заданной схеме, описывающей элементы интерфейса, а также данным, пришедшим в представление, создается каркас стра-

(дополняя необходимыми элементами) для формирования представления каждой отдельной страницы панели администратора.

Практическое применение модуля

Рассмотрим реализацию работы приложения с модулем генерации административного интерфейса. На первом этапе контроллер вызывает некоторое представление и передает в него данные из модели.

Пример реализации кода на языке программирования PHP:

```
// Метод контроллера: Редактировать frame
public function action_edit_crucial_frame($data=[])
{
    // Получим модель frame и данные для текущего элемента
```

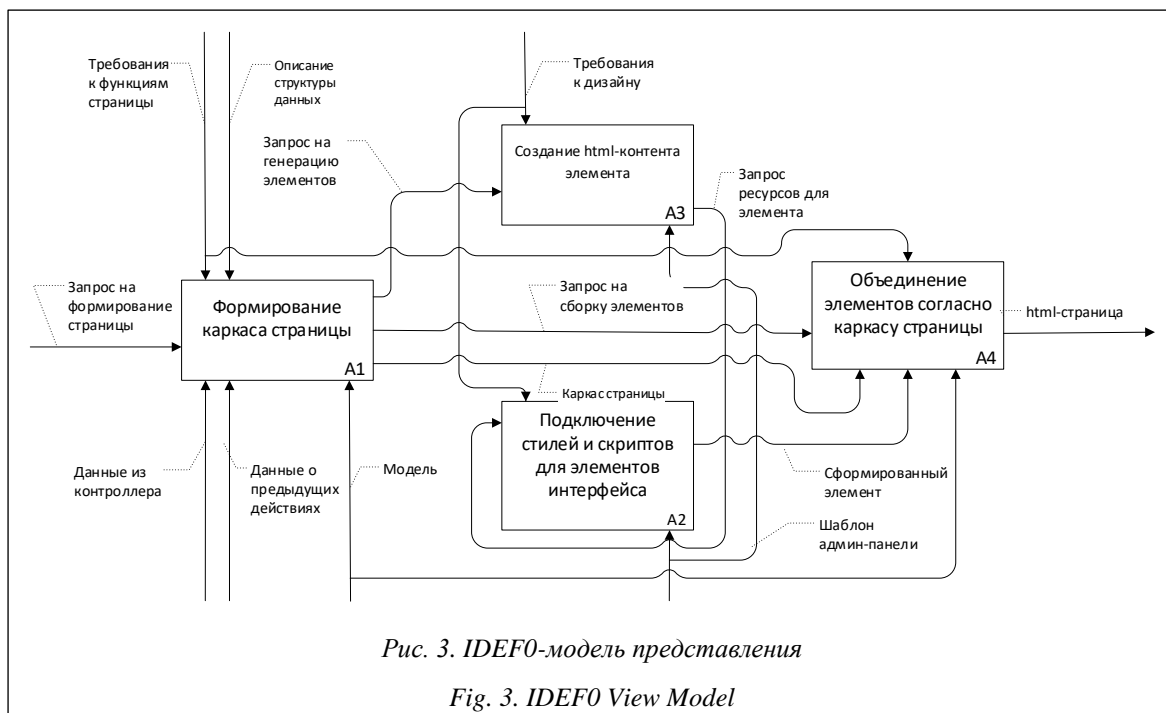


Рис. 3. IDEF0-модель представления

Fig. 3. IDEF0 View Model

```

    $FrameModel = $this->get-
    Model('FrameModel');
    if (!empty($data['id'])) {
        $Frame = $FrameModel->get-
        FrameById($data['id']);
    } else {
        $Frame = $FrameModel->get-
        FrameEmpty();
    };
    // Получим представление по
    имени
    $View = $this->getView('Frame-
    Model/CrucialFrameDetail');
    // Результат действия пользова-
    теля
    $arResult = [];
    // Если пользователь подтвердил
    внесение изменений
    if (isset($data['confirm'])) {
        // Передадим в нее данные от
        пользователя
        $arResult = $FrameModel-
        >editCrucialFrame($data);
    }
    // Сформируем новую страницу в
    качестве ответа
    $View->render([
        'Frame' => $Frame,
        'arResult' => $arResult,
    ]);
    }
    
```

В представлении формируем страницу админ-панели. Система позволяет добавить сквозные элементы (шапка, левое меню, подвал и т.п.) в родительский класс, а затем ис-

пользовать его как основу для конкретной страницы. Поэтому достаточно взять такую заготовку и добавить в нее необходимые элементы для редактирования пользователю.

Пример реализации кода на языке программирования PHP:

```

    // Представление: детальная страница
    фрейма
    class CrucialFrameDetail extends
    AdminPage
    {
        // Содержание основной части
        public function run()
        {
            // Создадим форму
            $FormMain = new \UI\Form([
                'action'=> \Samum\Helper\
                Router::getUrl(
                    'FrameModel',
                    'edit_crucial_frame',
                    [
                        'id' => $this->Frame-
                    >id,
                        'confirm' => true
                    ]),
            ]);
            // Добавим на форму заголовок и
            пару полей
            $FormMain
                ->setTitle('Фрейм')
                ->add(new \UI\Form\Input([
                    'name' => 'id',
                    'type' => 'hidden',
                    'value' => $this->Frame->id,
                ]))
                ->add(new \UI\Form\Input([
    
```

```

        'title' => 'Название',
        'name' => 'name',
        'value' => $this->Frame
->name,
    ]));
    // Создадим лист, на который
будем крепить элементы
    $BlankMain = new \UI\Page\
Blank();
    // Прикрепим форму на лист
    $BlankMain->add($FormMain);
    // Добавим лист в интерфейс
    $this->UI->add($BlankMain);
}

```

Теперь система сама сгенерирует страницу админ-панели на основе указанного макета (см. <http://www.swsys.ru/uploaded/image/2023-1/2023-1-dop/7.jpg>). Вид админ-панели зависит от шаблона, использованного для стилизации элементов.

На иллюстрации показан пример админ-панели с шаблоном Gentelella Admin. Это бесплатный шаблон административной панели, основанный на Bootstrap. Он содержит множество мощных плагинов и инструментов (в том числе с применением библиотеки jQuery) для реализации панелей под большинство типовых задач [16].

Выводы

В работе рассмотрены типовые компоненты, из которых состоит админ-панель. Описана структура созданного модуля, позволяющего генерировать рассмотренные ранее компоненты. Приведен пример использования разработанного модуля согласно описанной структуре. Программный продукт написан на языке PHP и предназначен для применения в веб-проектах на основе MVC-модели или фреймворках.

Основные преимущества разработанного продукта перед аналогичными модулями – это отсутствие привязки к конкретному фреймворку, возможность менять внешний вид админ-панели без изменения схемы страниц, отсутствие привязки к сайтам (созданная админ-панель может быть использована для управления любыми данными).

Дальнейшими направлениями исследования являются разработка набора виджетов, расширяющих функциональные возможности модуля, подготовка установочных пакетов для популярных веб-фреймворков, создание пакетов взаимозаменяемых тем оформления админ-панели.

Литература

1. Readyscript. Панель администратора. URL: https://readyscript.ru/manual/user_admin_panel.html (дата обращения: 19.06.2022).
2. Якимов А.С., Баженов Р.И. Разработка модуля страниц для панели администратора на Laravel // Постулат. 2017. № 1. С. 46.
3. Лучкин А.С. Модуль административной панели для личного кабинета абитуриента ГУАП: Свид. о регистр. ПрЭВМ № 2021682099. Рос. Федерация, 2021.
4. Карякин И.Ю., Галыгин Р.Е., Кунст К.Ю. Hi-Tech Dom – административная панель: Свид. о регистр. ПрЭВМ № 2020616361. Рос. Федерация, 2020.
5. Общество с ограниченной ответственностью «СМИ2». Административная панель информационной рекламно-обменной сети СМИ2: Свид. о регистр. ПрЭВМ № 2020614921. Рос. Федерация, 2020.
6. Усачев А.Ю. Динамическая генерация визуальных интерфейсов Flutter-приложений // Электронные библиотеки. 2020. Т. 23. № 5. С. 1093–1103. DOI: 10.26907/1562-5419-2020-23-5-1093-1103.
7. Оленников А.В. Разработка панели администратора сайта на фреймворке Yii2 // Молодые ученые в решении актуальных проблем науки: сб. матер. Всерос. науч.-практич. конф. студентов, аспирантов и молодых ученых. 2019. С. 510–512.
8. Галлини Н.И., Долинский А.П. Особенности разработки админ-панели для единого информационно-аналитического пространства образовательной организации высшего образования // Информационные системы и технологии в моделировании и управлении: сб. тр. V Междунар. науч.-практич. конф. 2020. С. 284–288.
9. Laravel Nova. Master Your Universe. URL: <https://nova.laravel.com/> (дата обращения: 08.11.2022).
10. Orchid. Build Admin Panels with Laravel Orchid. URL: <https://orchid.software/> (дата обращения: 08.11.2022).
11. Voyager. The Missing Laravel Admin. URL: <https://voyager.devdojo.com/> (дата обращения: 08.11.2022).
12. Bootstrap. Forms. <https://getbootstrap.com/docs/5.2/forms/overview/>. URL: <https://getbootstrap.com/docs/5.2/forms/overview/> (дата обращения: 19.06.2022).

13. Envato Market. Admin Dashboard Templates. URL: <https://themeforest.net/category/site-templates/admin-templates> (дата обращения: 19.06.2022).
14. Цифругл. Шаблон разработки ПО – Model View Controller (MVC). URL: <https://900913.ru/2021/05/18/software-design-pattern-model-view-controller-mvc/> (дата обращения: 19.06.2022).
15. Dart News & Updates. Method Cascades in Dart. URL: <https://news.dartlang.org/2012/02/method-cascades-in-dart-posted-by-gilad.html> (дата обращения: 19.06.2022).
16. GitHub. ColorlibHQ/gentelella. URL: <https://github.com/ColorlibHQ/gentelella> (дата обращения: 19.06.2022).

Software & Systems
DOI: 10.15827/0236-235X.141.175-182

Received 25.08.22, Revised 09.11.22
2023, vol. 36, no. 1, pp. 175–182

A control panel generation module for web projects

*S.V. Zhukov*¹, Postgraduate Student, coder.stas@gmail.com

*A.A. Suslin*¹, Student, andriw_suslin@mail.ru

*O.A. Kovaleva*¹, Dr.Sc. (Engineering), Associate Professor, Professor of the Department of Mathematical Modeling and Information Technologies, solomina-oa@yandex.ru

¹ Tambov State University of G.R. Derzhavin, Tambov, 392036, Russian Federation

Abstract. The article considers the problems of creating a control panel for web projects. It describes the implementation of a program module written in the php programming language, which allows creating an admin panel for web resources. The module does not require skills in HTML template layout due to the Gentelella Admin template preinstalled in the module (developed and maintained by Colorlib and AigarsSilkals). The described module can be used both in projects built on the basis of the popular MVC model and in resources written on the basis of the PHP framework. The control panel includes many standard controls (input fields, switches, tables, lists, etc.), and can also support custom components that add new controls.

The purpose of the work is to optimize the process of developing web projects using the developed module. The subject of the research are ways to implement the administrative panel of a web application. The presented module allows generating html, css and js code for visualizing the control panel. The user interface is set in the form of php code. The objects created in it are interface elements that grouped into forms and panels that a developer needs. Using the module allows reducing the development time spent on the forming the control panel, and also allows creating applications without skills in layout and organization of the client side of the user interface.

The theoretical significance of the work is in the analysis of typical elements of the admin panel, the description of the principles of operation of the created module for generating the admin panel. The practical significance is to reduce the development time of web projects using the control panel generation module.

Keywords: Web project, administrative panel, web project control panel, interface generation, components, MVC model, CMS, framework.

References

1. Readyscript. Admin Panel. Available at: https://readyscript.ru/manual/user_admin_panel.html (accessed June 19, 2022) (in Russ.).
2. Yakimov A.S., Bazhenov R.I. Develop module of pages for admin panel on Laravel. *Postulat*, 2017, no. 1, 46 p. (in Russ.).
3. Luchkin A.S. *Administrative Panel Module for the Personal Account of the SUAI Applicant*. Patent RF, no. 2021682099, 2021 (in Russ.).
4. Karyakin I.Yu., Galygin R.E., Kunst K.Yu. *Hi-Tech Dom – Administrative Panel*. Patent RF, no. 2020616361, 2020 (in Russ.).
5. Limited Liability Company "SMI2". *Administrative Panel of the Information Advertising and Exchange Network of Media2*. Patent RF, no. 2020614921, 2020 (in Russ.).
6. Usachev A.Yu. Dynamically generated Ui on Flutter. *Russian Digital Libraries J.*, 2020, vol. 23, no. 5, pp. 1093–1103. DOI: 10.26907/1562-5419-2020-23-5-1093-1103 (in Russ.).

7. Olennikov A.V. Development of the site administrator panel on the Yii2 framework. *Proc. All-Russ. Sci. and Pract. Conf. Young Scientists in Solving Actual Problems of Science*, 2019, pp. 510–512 (in Russ.).
8. Gallini N.I., Dolinskiy A.P. Features of the development of an admin panel for a single information and analytical space of an educational organization of higher education. *Proc. V Int. Conf. Information Systems and Technologies in Modeling and Management*, 2020, pp. 284–288 (in Russ.).
9. *Laravel Nova. Master Your Universe*. Available at: <https://nova.laravel.com/> (accessed November 08, 2022).
10. *Orchid. Build Admin Panels with Laravel Orchard*. Available at: <https://orchid.software/> (accessed November 08, 2022).
11. *Voyager. The Missing Laravel Admin*. Available at: <https://voyager.devdojo.com/> (accessed November 08, 2022).
12. *Bootstrap. Forms*. Available at: <https://getbootstrap.com/docs/5.2/forms/overview/> (accessed June 19, 2022).
13. *Envato Market. Admin Dashboard Templates*. Available at: <https://themeforest.net/category/site-templates/admin-templates> (accessed June 19, 2022) (in Russ.).
14. *Tsifrugl. Software Development Template – Model View Controller (MVC)*. Available at: <https://900913.ru/2021/05/18/software-design-pattern-model-view-controller-mvc/> (accessed June 19, 2022) (in Russ.).
15. *Dart News & Updates. Method Cascades in Dart*. Available at: <https://news.dartlang.org/2012/02/method-cascades-in-dart-posted-by-gilad.html> (accessed June 19, 2022).
16. *GitHub. ColorlibHQ/gentelella*. Available at: <https://github.com/ColorlibHQ/gentelella> (accessed June 19, 2022).

Для цитирования

Жуков С.В., Суслин А.А., Ковалева О.А. Модуль генерации панели управления для веб-проектов // Программные продукты и системы. 2023. Т. 36. № 1. С. 175–182. DOI: 10.15827/0236-235X.141.175-182.

For citation

Zhukov S.V., Suslin A.A., Kovaleva O.A. A control panel generation module for web projects. *Software & Systems*, 2023, vol. 36, no. 1, pp. 175–182 (in Russ.). DOI: 10.15827/0236-235X.141.175-182.