

## Проектирование LC/NC платформы на базе фреймворка Laravel

*Г.С. Мизюков*

*Ростовский государственный университет путей сообщения, Ростов-на-Дону*

**Аннотация:** В статье освещаются отличительные признаки low-code, no-code платформ, их преимущества и недостатки, делается обзор существующих исследований в области low-code, no-code разработки, рассматриваются основные компоненты проектируемой платформы и фрагменты алгоритмов, отвечающих за генерацию программного кода.

**Ключевые слова:** low-code, no-code, разработка программного обеспечения, laravel, проектирование платформ

### Введение

Развитие информационных технологий способствует появлению новых подходов к проектированию и разработке программного обеспечения. Одними из таких подходов являются low-code (LC) и no-code (NC). Оба подхода предполагают разработку программного обеспечения без написания программного кода. Отличие состоит лишь в том, что low-code подход допускает написание и редактирование программного кода в небольшом объеме (например, написание sql-запроса). Платформы, разработанные на основе данных подходов, позволяют снизить затраты за счет сокращения времени разработки программного продукта, а также требований к квалификации персонала, и ускорить процесс цифровой трансформации организации за счет использования готовых шаблонов и решений. Основными отличительными признаками LC/NC платформ являются наличие визуального интерфейса с поддержкой drag and drop функционала, наличие готовых шаблонов, компонентов, модулей и решений, а также поддержка быстрого развертывания программного продукта. Среди недостатков LC/NC платформ следует отметить зависимость от платформы и сложность миграции на другие сервисы, ограниченность в функционале (особенно при работе с NC платформами), наличие персонала с минимальными знаниями программирования (при работе с LC

---



платформами). Однако, несмотря на недостатки, популярность данных подходов с каждым годом возрастает, чему свидетельствует обилие исследований по данной теме.

### **Предшествующие работы**

Возможность сокращения цикла проектирования программного обеспечения, разработка программ без написания программного кода являются актуальными направлениями, исследованиям которых уделяется большое внимание. Так, например, в статье [1] автор рассматривает наиболее популярные low-code и no-code платформы (Salesforce, OutSystem, Microsoft Power App), описывает основные отличия между ними и некоторые из проблем, с которыми можно столкнуться при переходе к данным платформам. Авторы статей сосредоточили свои исследования вокруг low-code платформ: сделали обзор платформ [2, 3], показали их преимущества [4, 5], в качестве примеров привели свои наработки в области low-code [6]. В статье [7] авторы освещают основные вызовы, с которыми можно столкнуться при разработке low-code и no-code платформ. Статья [8] посвящена обзору вспомогательных инструментов, основанных на машинном обучении, с помощью которых возможна генерация небольших фрагментов программного кода. В статье [9] авторы описывают критерии обучающей выборки для тренировки нейронной сети, которая способна проводить анализ исходного кода программ. В статье [10] авторы предлагают no-code платформу на основе предсказательной аналитики для анализа социальных сетей.

### **Описание LC/NC платформы на базе фреймворка Laravel**

Для проектирования и дальнейшей реализации LC/NC платформы был выбран фреймворк Laravel. Данный фреймворк позволяет разрабатывать веб-

---

приложение любой сложности. В основе фреймворка лежит архитектура MVC (Model-View-Controller), которая позволяет гибко управлять структурой веб-приложения. На рис. 1 показан процесс генерации кода  $n$ -ой веб-страницы. Процесс начинается с получения запроса от пользователя на показ  $n$ -ой веб-страницы. Далее запрос получает маршрутизатор (*Router*). Маршрутизатор определяет тип запроса, его параметры и передает их в контроллер (*Controller*). В контроллере определен ряд методов, вызов, которых определяется параметрами запроса. Таким образом, на основании запроса определяется  $n$ -й метод из контроллера. Затем контроллер запрашивает данные из базы данных на основе модели (*Model*) объекта. Под моделью объекта понимается таблица из базы данных. После этого контроллер передает данные в размеченный шаблон (*View*) и возвращает пользователю готовую веб-страницу.

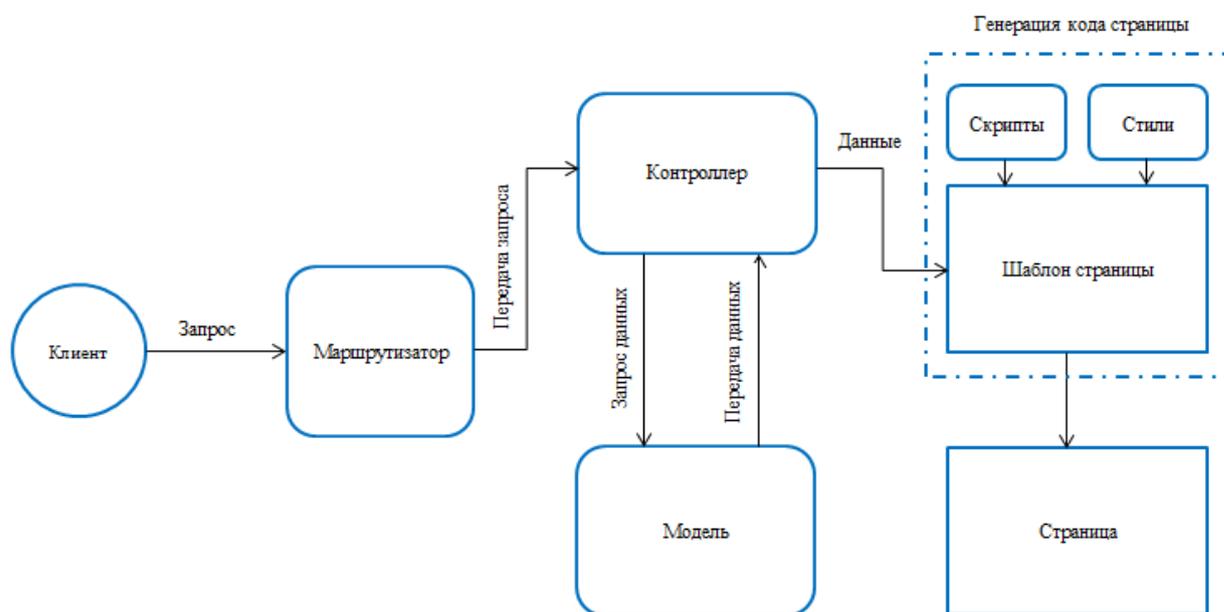


Рис. 1. – Модель процесса генерации кода веб-страницы

Далее рассмотрим реализацию генерации кода  $n$ -ой веб-страницы на фреймворке Laravel. На первом этапе необходимо определить модели объектов базы данных. В качестве основных моделей будут выступать 4 таблицы: *pages*, *sections*, *elements*, *element\_contens* (рис. 2). Таблица *pages*

хранит информации о заголовке страницы, её *url* адрес, а также сведения, необходимые для индексации страницы поисковыми системами. Таблица *sections* отвечает за хранение логической структуры веб-страницы, в таблице *elements* хранятся объекты интерфейса веб-страницы, такие, как кнопка (`<button>`, `<input type="submit" value="button">`, `<input type="radio">`), текстовое поле (`<input type="text">`, `<textarea>`), таблица (`<table>`) и т.д. Таблица *element\_contents* отвечает за хранение содержимого объектов из таблицы *elements*.

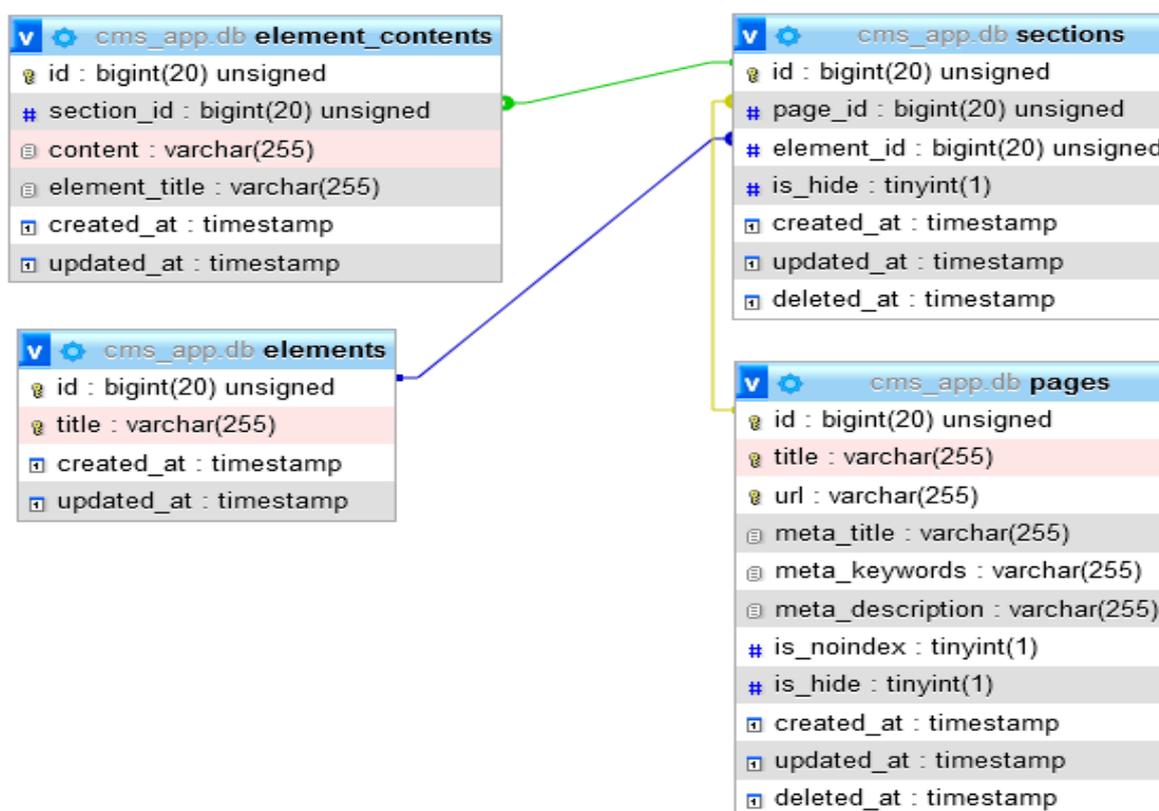


Рис. 2. – Фрагмент физической модели базы данных с отражением связей между таблицами

После создания этих таблиц необходимо создать универсальное правило для поступающих запросов в маршрутизатор (рис. 3) и контроллер, который будет отвечать за обработку этих запросов (рис. 4).

```
Route::get('/{slug}', [UrlController::class, 'index']->name('url'));
```

Рис. 3. – Фрагмент программного кода маршрутизатора (файл *web.php*)

Контроллер (*UrlController*) на основе поступившего запроса осуществляет поиск информации в модели *Page*. Если информация в модели *Page* найдена, то осуществляется выборка элементов страницы и их содержимого. Затем полученные данные передаются в шаблон, после чего пользователю возвращается готовая веб-страница. Если модель *Page* вернула *null*, то пользователю вернется страница 404.

```
public function index($slug)
{
    $contents = array();
    $page = Page::where('slug', '=', $slug)->firstOrFail();

    foreach($page->sections as $item)
    {
        $element = Element::select('title')->findOrFail($item->element_id);
        $data = ElementContent::where('section_id', '=', $item->id)
            ->where('element_title', '=', $element->title)->firstOrFail();

        array_push($contents, ['data' => $data]);
    }

    $view = ($page != null) ? 'templates.base-template' :
        'templates.404-template';

    return view($view, ['contents' => $contents]);
}
```

Рис. 4. – Фрагмент программного кода контроллера (файл *UrlController.php*)

### Заключение

На основе небольшого примера, разработанного на фреймворке Laravel, продемонстрировано создание веб-приложения, которое способно

генерировать веб-страницы с динамичным содержимым. Структура такого приложения позволяет создавать простые веб-ресурсы без написания программного кода, ориентированные на малый и средний бизнес. Среди минусов стоит отметить отсутствие сложной бизнес-логики. Но, несмотря на это, данное веб-приложение может занять свой сегмент рынка программных продуктов, на базе которого возможно построение полноценной LC/NC платформы.

### Литература

1. Магомадов В.С. Платформы low-code и no-code как способ сделать программирование более доступным для широкой общественности // МНИЖ. 2021. №6-1 (108). С.100-103.
  2. Гаврилина Дарья Эдуардовна, Манцивода Андрей Валерьевич Low-code и объектные электронные таблицы // Известия Иркутского государственного университета. Серия: Математика. 2022. №.40. С. 93-103.
  3. Яковлев Г.С., Иванов Ф.Ф. Использование low-code платформ при переходе на процессный подход в создании автоматизированных систем // Вест. КРАУНЦ. Физ.-мат. науки. 2020. №1. С. 120-126.
  4. Bock, A.C., Frank, U. Low-Code Platform. *Bus Inf Syst Eng* 63. 2021. Pp. 733–740. URL: [doi.org/10.1007/s12599-021-00726-8](https://doi.org/10.1007/s12599-021-00726-8)
  5. Phalake, V.S., Joshi, S.D. Low Code Development Platform for Digital Transformation. In: Kaiser, M.S., Xie, J., Rathore, V.S. (eds) *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*. Lecture Notes in Networks and Systems. 2021. Vol. 190. Springer, Singapore. URL: [doi.org/10.1007/978-981-16-0882-7\\_61](https://doi.org/10.1007/978-981-16-0882-7_61)
  6. Bucaioni, A., Cicchetti, A. & Ciccozzi, F. Modelling in low-code development: a multi-vocal systematic review. *Softw Syst Model* 21. 2022. Pp. 1959–1981. URL: [doi.org/10.1007/s10270-021-00964-0](https://doi.org/10.1007/s10270-021-00964-0)
-

7. Rokis, K., Kirikova, M. Challenges of Low-Code/No-Code Software Development: A Literature Review. In: Nazaruka, Ė., Sandkuhl, K., Seigerroth, U. (eds) Perspectives in Business Informatics Research. BIR 2022. Lecture Notes in Business Information Processing. 2022. Vol. 462. Springer, Cham. URL: [doi.org/10.1007/978-3-031-16947-2\\_1](https://doi.org/10.1007/978-3-031-16947-2_1)

8. Какутин Д.Ю., Дмитриев А.С., Абрамов И.М. Обзор вспомогательных инструментов на основе машинного обучения для написания исходного кода программ // Инженерный вестник Дона, 2022, №5. URL: [ivdon.ru/ru/magazine/archive/n5y2022/7675](https://ivdon.ru/ru/magazine/archive/n5y2022/7675)

9. Какутин Д.Ю., Дмитриев А.С. Формирование и анализ эффективности выборки для обучения языковых моделей распознаванию и анализу исходного кода программ // Инженерный вестник Дона, 2022, №5. URL: [ivdon.ru/ru/magazine/archive/n5y2022/7682](https://ivdon.ru/ru/magazine/archive/n5y2022/7682)

10. Schötteler, S., Laumer, S., Schuhbauer, H., Scheidthauer, N., Seeberger, P., Miethsam, B. A No-Code Platform for Tie Prediction Analysis in Social Media Networks. In: Ahlemann, F., Schütte, R., Stieglitz, S. (eds) Innovation Through Information Systems. WI 2021. Lecture Notes in Information Systems and Organisation. 2021. Vol. 47. Springer, Cham. URL: [doi.org/10.1007/978-3-030-86797-3\\_32](https://doi.org/10.1007/978-3-030-86797-3_32)

### References

1. Magomadov V.S. MNIJ. 2021. №6-1 (108). Pp.100-103.
  2. Gavrilina Darya Eduardovna, Mantsivoda Andrey Valerevich. Izvestiya Irkutskogo gosudarstvennogo universiteta. Seriya: Matematika. 2022. №40. Pp. 93-103.
  3. Yakovlev G.S., Ivanov F.F. Vest. KRAUNTS. Fiz.-mat. nauki. 2020. №1. Pp. 120-126.
  4. Bock, A.C., Frank, U. Bus Inf Syst Eng 63, 2021. Pp. 733–740. URL: [doi.org/10.1007/s12599-021-00726-8](https://doi.org/10.1007/s12599-021-00726-8)
-



5. Phalake, V.S., Joshi, S.D. In: Kaiser, M.S., Xie, J., Rathore, V.S. (eds) Information and Communication Technology for Competitive Strategies (ICTCS 2020). Lecture Notes in Networks and Systems. 2021. Vol. 190. Springer, Singapore. URL: [doi.org/10.1007/978-981-16-0882-7\\_61](https://doi.org/10.1007/978-981-16-0882-7_61)
6. Bucaioni, A., Cicchetti, A. & Ciccozzi, F. *Softw Syst Model* 21, 2022. Pp. 1959–1981. URL: [doi.org/10.1007/s10270-021-00964-0](https://doi.org/10.1007/s10270-021-00964-0)
7. Rokis, K., Kirikova, M. In: Nazaruka, Ę., Sandkuhl, K., Seigerroth, U. (eds) Perspectives in Business Informatics Research. BIR 2022. Lecture Notes in Business Information Processing. 2022. Vol. 462. Springer, Cham. URL: [doi.org/10.1007/978-3-031-16947-2\\_1](https://doi.org/10.1007/978-3-031-16947-2_1)
8. Kakutin D.YU., Dmitriev A.S., Abramov I.M. *Inzhenernyj vestnik Dona*, 2022, №5. URL: [ivdon.ru/ru/magazine/archive/n5y2022/7675](https://ivdon.ru/ru/magazine/archive/n5y2022/7675)
9. Kakutin D.YU., Dmitriev A.S. *Inzhenernyj vestnik Dona*, 2022, №5. URL: [ivdon.ru/ru/magazine/archive/n5y2022/7682](https://ivdon.ru/ru/magazine/archive/n5y2022/7682)
10. Schötteler, S., Laumer, S., Schuhbauer, H., Scheidthauer, N., Seeberger, P., Miethsam, B. In: Ahlemann, F., Schütte, R., Stieglitz, S. (eds) Innovation Through Information Systems. WI 2021. Lecture Notes in Information Systems and Organisation. 2021. Vol. 47. Springer, Cham. URL: [doi.org/10.1007/978-3-030-86797-3\\_32](https://doi.org/10.1007/978-3-030-86797-3_32)