Geekbrains

**Разработка интернет-магазина с использованием Laravel + DDD (Domain-Driven Design)**

Программа: Разработчик

Специализация: Fullstack разработчик

Баглаев А.Н.

Москва

2025

**Глава 1. Теоретические основы разработки интернет-магазина…………………….3**

**1.1. Современные тенденции электронной коммерции……………………………….3**

* Рынок электронной коммерции: статистика и перспективы
* Преимущества интернет-магазинов перед традиционной торговлей
* Основные требования к интернет-магазинам

**1.2. Обзор технологий для разработки веб-приложений……………………………18**

* Серверные языки программирования (PHP, Python, Node.js)
* Фреймворки для backend-разработки (Laravel, Django, Express.js)
* Frontend-технологии (HTML/CSS, JavaScript, Vue.js/React)
* Системы управления базами данных (MySQL, PostgreSQL)

**1.3. Особенности фреймворка Laravel…………………………………………………42**

* История и преимущества Laravel
* Архитектура MVC в Laravel
* Встроенные инструменты (Eloquent ORM, Blade, Artisan, миграции)
* Безопасность в Laravel (CSRF-защита, валидация, аутентификация)

**Глава 2. Практическая часть. Разработка интернет-магазина с использованием DDD. ……………………………………………………………………………………….56**

2.1. Архитектура проекта

2.2. Реализация доменной логики (на примере модуля Catalog) ……………………….56

2.3. Работа с корзиной (модуль Cart) ……………………………………………………64

2.4. Оформление заказа (модуль Order) …………………………………………………..67

**Заключение …………………………………………………………………………….. 74**

**Глава 1. Теоретические основы разработки интернет- магазина**

* 1. **Современные тенденции электронной коммерции**
     1. Рынок электронной коммерции: статистика и перспективы.

Интернет-торговля включает в себя розничную или оптовую продажу товаров, которая осуществляется по интернет-заказам покупателей, поступающим продавцам с сайтов. При этом условия продажи товаров (цены, доставка и др.) размещены на интернет-ресурсах продавцов или торговых онлайн-площадках.

Рынок интернет-торговли в России по ряду причин отстаёт в развитии от рынков онлайн-торговли ряда зарубежных стран с крупными экономиками и высоким уровнем жизни населения. Для российского интернет-бизнеса это создаёт уникальную возможность оперативно корректировать свою деятельность в этой сфере с учётом зарубежного опыта и минимизировать свои издержки на пути расширения бизнеса.

Электронная коммерция родилась в 1960 году в США, когда компании American Airlines и IBM запустили бизнес-систему по отслеживанию и заказу мест на авиарейсах (билеты доставлялись по почте, либо их можно было получить непосредственно в аэропорту). Надо отметить, что на разработку системы была затрачена огромная по тем временам сумма в размере $150 млн, на которую можно было приобрести воздушный флот из нескольких сотен пассажирских самолётов.

В 80-е годы XX века e-commerce уже проникает во многие сферы американской экономики, но бурное развитие этой отрасли началось после массового внедрения интернета в экономику и бизнес-процессы. С начала 90-х годов управление интернетом переходит из‑под контроля американского правительства в руки частных компаний. В 1992 году конгресс США дал «добро» на включение в орбиту e-commerce интернет-сетей, а через год закладывается технологическая база для использования электронных денег, что позволило уже в 1994 году создать первую в мире электронную платёжную систему.

Официальная дата рождения интернет-торговли – 11 августа 1994 года. На следующий день об «открытии интернета» сообщила в своей статье New York Times, рассказав о том, что американец Фил Бранденбергер приобрёл на веб-сайте за $12,48 (без учёта стоимости доставки) новый альбом Стинга Ten Summoners' Tales и оплатил покупку со своей банковской карты. Так что этим летом можно отмечать 30-летие начала интернет-торговли на планете Земля.

В 1995 году была создана Amazon, ставшая первой компанией, специализирующейся в сфере электронной коммерции. В настоящее время доля этого крупнейшего маркетплейса на рынке e-commerce США оценивается в районе 40%. В этом же году начинает обслуживать интернет-клиентов и онлайн-банк Bank of Amerika.

В России по понятным причинам (чего стоят одни «лихие 90-е») процесс интернет-коммерциализации шёл медленнее. В 2001 году вводится в оборот первая российская кредитная карточка, являющаяся основным инструментом интернет-платежей: компания STB-Card выпустила международную кредитную карту «Путникъ» для оплаты товаров и услуг на территории РФ и СНГ.

Уже через двенадцать лет объём e-commerce в РФ достиг уровня в 240 млрд руб., а в течение четырёх последующих лет (по итогам 2017 года) он превысил порог в 1 трлн руб. — это более $17 млрд в пересчёте по среднегодовому курсу доллара (58,35 руб. за $1). Для сравнения: в 2017 году мировой объём e-commerce в B2B-сегменте составил $7,66 трлн, а в B2C-сегменте – $2,14 трлн.

**Подъём интернет-торговли и пандемия короновируса COVID-19.**

По данным исследования, проведённого в 2020 году аналитической компании Data Insight, в период 2011–2019 гг.онлайн-торговля в РФ ежегодно росла более чем на 20%. За эти 9 лет объём покупок интернет-товаров россиянами увеличился до 1,72 трлн руб. (с учётом НДС).

В 2018 наметились все предпосылки для резкого увеличения темпов роста рынка e-commerce. Одним из ключевых факторов изменений на рынке стали значительные вложения в рекламу (как в телевизионную, так и в веб-пространстве) наиболее крупных игроков рынка e-commerce, а также инвестиции в развитие инфраструктуры онлайн-торговли, прежде всего – логистической. Новые покупатели, привлечённые в III квартале 2018 года рекламой и расширением территории, охваченной доставкой интернет-товаров, и начавшие покупать товары в онлайн-формате, вместе обеспечили в течение следующего года денежные поступления продавцам в сегменте e-commerce на сумму свыше 100 млрд руб., удвоив (а то и утроив) доходы от торговли крупнейших субъектов российского онлайн-рынка.

В качестве мощного катализатора подъёма онлайн-торговли выступила пандемия коронавируса COVID-19. Например, оборот веб-торговли в Подмосковье в пиковый период коронакризиса во II квартале 2020 года превысил «доковидный» уровень в 12 раз! Рост российского онлайн-рынка по итогам 2020 года составил рекордные 44%, а его объём вышел на уровень 2,5 трлн руб.

Интересен прогноз аналитиков агентства Data Insight по развитию торговли в сегменте e-commerce РФ на 2020–2024 гг., сделанный с учётом влияния коронакризиса на увеличение онлайн-продаж – см. диаграмму, представленную выше. По их оценкам, в эти 5 лет совокупный среднегодовой темп роста онлайн-торговли (CAGR) должен был превысить 33% и привести к увеличению общего объёма продаж интернет-товаров более чем в 4 раза – до 7,2 трлн руб.

Эксперты оценили добавочный рост продаж на онлайн-рынке России в этот 5-летний период за счёт влияния прошедшей пандемии на уровне 1,6 трлн руб. Среди позитивных и негативных факторов, повлиявших на динамику дополнительного роста интернет-продаж:

* нахождение в карантине более 10 млн россиян, значительная часть из которых стала «новыми покупателями» интернет-торговли;
* перевод на удалённый режим работы в РФ порядка 7 млн чел., которые также пополнили ряды покупателей на рынке e-commerce;
* большинство из пребывающих в карантинном режиме оценили «плюсы» покупки и доставки «на дом» онлайн-товаров, продолжая и после завершения коронакризиса пользоваться услугами интернет-торговли;
* повышение спроса в период карантина, приведшее к созданию высокой базы, должно было сказаться в 2021 году, линейно уменьшая темпы роста онлайн-торговли;
* падение активности потребителей и покупательной способности, прежде всего в сегменте премиальных товаров, потребляемых средним классом.

Аналитики Data Insight, делая прогноз, основывались на предположении, что фактор «карантинного» приобщения к интернет-торговле большого количества новых покупателей будет значимым в 2020–2021 гг., а потом перестанет работать. С другой стороны, в своих расчётах они опирались на то, что росту e-commerce в постпандемийный период будет способствовать и «каннибализация» покупок в оффлайн-магазинах вошедшими во вкус онлайн-покупателями.

Кроме того, предполагалось, что и после коронакризиса продолжится рост продаж в онлайне FMCG (Fast-Moving Consumer Goods — товары «частого применения», т. е. повседневного спроса, например, продукты питания). В этой категории товаров, на которую приходится свыше половины розничной торговли в РФ, до настоящего времени доля онлайн-продаж ниже, чем в других сегментах.

**Катализаторы роста онлайн-торговли в РФ.**

Прогноз экспертов Data Insight, сделанный 5 лет назад, оказался немного заниженным — реальные итоги развития российского рынка e-commerce в последующие годы превзошли все ожидания и показали, что у интернет-торговли в России большой потенциал для роста. Тем более, что появились новые факторы, стимулирующие развитие торговых онлайн-площадок.

Например, аналитики, не могли знать, что в РФ государство начнёт активно поддерживать самозанятых и предпринимателей из сегмента малого и среднего бизнеса, реализующих на маркетплейсах товары/услуги собственного производства, а также продукты питания и напитки.

Так, в Московской области в 2021 году была запущена программа субсидирования такой деятельности по упрощённой схеме (заявления направляются в электронном формате с минимальным количеством подтверждающих документов). Размер субсидии — не более 1/2 от общего объёма затрат по размещению, продвижению и реализации товаров или услуг на отобранных Мининвестом региона торговых онлайн-площадках, при этом её максимальная сумма — полмиллиона рублей. В число первых отобранных к участию в программе вошли 8 торговых интернет-площадок, в т.ч. Wildberries, Ozon, Авито, Яндекс. Маркет, Яндекс. Еда. Программа нашла большой отклик у самозанятых и субъектов малого и среднего предпринимательства и продолжает успешно работать до сих пор.

Одним из главных факторов, повлиявших на скакчкообразный рост интернет-торговли в России, стало соревнование регионов РФ в привлечении инвестиций, усилившееся в последние годы. Создавая новые распределительно-логистические центры в субъектах Российской Федерации, крупные игроки рынка получают ряд федеральных и региональных льгот (имущественных, земельных, налоговых, таможенных и др.), возможность льготного кредитования, использования субсидий и т. д. Это способствовало лавинообразному росту логистической инфраструктуры крупнейших российских маркетплейсов.

Такая тенденция будет продолжаться и дальше. Например, Wildberries, построит в Ярославской области распределительный комплекс площадью около 200 тыс. кв. метров, стоимость которого составит 10 млрд руб. Компания планирует выделить 7,8 млрд руб. на создание распределительно-логистического центра площадью более чем 150 тыс. кв. метров на территории ОЭЗ «Оренбуржье». Инвестиции в размере 8,5 млрд руб. Wildberries направит в строительство распределительного комплекса в Пермском крае площадью 170 тыс. кв. метров. В настоящее время у этого крупнейшего российского маркетплейса уже больше 30 тыс. пунктов выдачи заказов в регионах РФ, а общая площадь распределительно-логистических центров компании достигла 2,9 млн кв. метров, в планах Wildberries – её удвоение.

**2020–2023 годы: результаты превзошли ожидания**

Объём рынка интернет-торговли в РФ в «ковидном» 2020 году на полтриллиона рублей превысил прогноз агентства и составил 2,7 трлн руб. При этом на крупных игроков рынка (Wildberries, Яндекс. Маркет, Ozon, AliExpress Россия — и это при других действующих 35 тыс. российских интернет-продавцах) пришлась около 50% продаж, их суммарный оборот вырос на 108%, достигнув 721 мрлд руб., а средний чек этих торговых интернет-площадок составил 1,78 тыс.руб.

Рост остального интернет-рынка составил 38%, его объём – 1,7 трлн руб., а средний чек — превысил 6,2 тыс.руб. Последнее обстоятельство говорит о том, что крупные покупки (например, бытовой техники и электроники) покупатели предпочитали делать в специализированных интернет-магазинах.

Доля онлайн-торговли выросла в 2020 году в полтора раза, выйдя на уровень 9% от всего объёма ритейла (на 3% больше, чем годом ранее). Заметно вырос процент интернет-продаж товаров непродовольственной группы — до 21%, в этом сегменте в 2020 году оффлайн-торговля потеряла 7% рынка.

В 2021 году, по данным Ассоциация компаний интернет-торговли, объём онлайн-торговли в РФ увеличился на 13% и составил 3,6 трлн руб. При этом на долю e-commerce по итогам года пришлось 9,2% от общего объёма розничной торговли, а количество интернет-покупателей превысило порог в 70 млн человек.

В 2022 году объём всей российской интернет-экономики достиг 12,2 трлн руб. (рост по сравнению с 2021 годом на 29%), а рынка розничной онлайн-торговли – 5,7 трлн руб. (с количеством заказов 2,8 млрд). 2022 год стал для рынка e-commerce России одним из лучших по динамике роста количества заказов: прирост на 65%.

И это в условиях ухода с российского рынка ряда европейских брендов, на которые приходилась достаточно большая доля рынка, и значительного ослабления премиального сегмента. Однако быстрое замещение ушедших брендов и параллельный импорт позволили стабилизировать ситуацию и поддержать товарооборот.

Окончательные итоги работы российского рынка e-commerce в 2024 году ещё не подведены. По прогнозу Data Insigh, доля онлайн-торговли на отечественном рынке ретейла и в этом году вырастет — до 19% от общего объёма розничной торговли. При этом объём продаж составит 7,9 трлн руб., увеличившись на 38%.

По оценке Российской ассоциации электронных коммуникаций, общий объём рынка e-commerce в РФ по итогам прошедшего года взлетит до 15,4 трлн руб. (рост в годовом измерении на уровне 35%). Но сюда входят не только продажи товаров, но и, например, реализация авиабилетов, услуг в туротрасли, продажи в общепите и т. п. Без учёта этих сегментов, по данным «Infoline-аналитики», объём продаж на рынке интернет-торговли России по итогам 2025 года составит 8,2 трлн руб., а по сравнению с предыдущим годом он увеличится на 44%.

**Тренды мировой интернет-торговли**

А в каком направлении в этот период двигался мировой рынок онлайн-торговли? Авторитетное международное агентство Wunderman Thompson в середине 2023 года провело масштабное исследование трендов в интернет-торговле, охватившее ряд государств Европы, Азии, Америки и Африки. В исследовании участвовали более 31,6 тыс. респондентов в 18 странах мира.

Анализ результатов опросов респондентов показал, что, если 2022 году интернет-покупки делали 57%, то в 2023 году этот процент увеличился до 58%, а в течение ближайших 10 лет онлайн-торговлей будет охвачено более 2/3 жителей Земли. При этом 68% опрошенных отметили, что интернет-шоппинг стал комфортнее.

Результаты исследования подтвердили, что онлайн-торговля оправилась от последствий пандемии коронавируса COVID-19.

При этом распространение работы в удалённом режиме значительно изменило поведение покупателей: «удалённые» работники значительно чаще покупают в интернете. Практически 2/3 (65%) респондентов говорят об этом. Примерно столько же опрошенных потребителей (66%) отмечают, что, работая дома, они больше времени тратят на изучение товаров, реализуемых онлайн.

При этом распространение работы в удалённом режиме значительно изменило поведение покупателей: «удалённые» работники значительно чаще покупают в интернете. Практически 2/3 (65%) респондентов говорят об этом. Примерно столько же опрошенных потребителей (66%) отмечают, что, работая дома, они больше времени тратят на изучение товаров, реализуемых онлайн.

В то же время покупателям нравится наличие у продавцов разных каналов продаж. Маркетплейсы доминируют на онлайн-рынке, но их доля в интернет-продажах стабилизировалась и практически не растёт. А вот прямые онлайн-продажи, минуя посредников, растут, но этот фактор работает только для компаний, умеющих выстраивать эффективные каналы коммуникации с потребителями, создавая конкуренцию крупным торговым интернет-площадкам с использованием разнообразных интернет-ресурсов — от веб-сайтов до соцсетей.

Для покупателей на первом месте по приоритетам находится привлекательная цена товара: так считает 56% опрошенных. Более половины респондентов (53%) готовы не покупать на маркетплейсах, а приобретать необходимые товары по более выгодным ценам у производителей, почти половина опрошенных (46%) при выборе места покупки жёстко ориентируются именно на цену товара.

Покупателей также привлекают качественный онлайн-сервис, включая оперативную доставку, возможность бесплатного возврата товаров, гарантийное и сервисное обслуживание. При этом предпочтение отдаётся брендам, которые ассоциируются с этичным и экологичным бизнесом.

Возглавляет топ стран-лидеров по онлайн-покупкам Мексика (69% от общего объёма затрат мексиканцев приходится на интернет-торговлю). Далее идёт Колумбия (66%), и только третье место занимает представитель стран «Большой семёрки» – Франция с показателем 48%.

Один из главных итогов исследования: 71% потребителей ищут подходящие товары, используя только интернет. Этот факт красноречиво говорит о том, что онлайн-продавцам необходимо создавать и развивать собственные интернет-каналы реализации предлагаемых ими товаров.

**Интернет-торговля в России**

В России пока доля интернет-торговли в общем объёме ритейла не так велика, но с каждым годом она растёт. На диаграммах ниже представлены результаты исследования Центра конъюнктурных исследований Института статистических исследований и экономики знаний НИУ ВШЭ, проведённого в 2024 году на базе ежеквартальных опросов Росстатом с охватом более 10 тыс. торговых предприятий из 82 субъектов РФ.

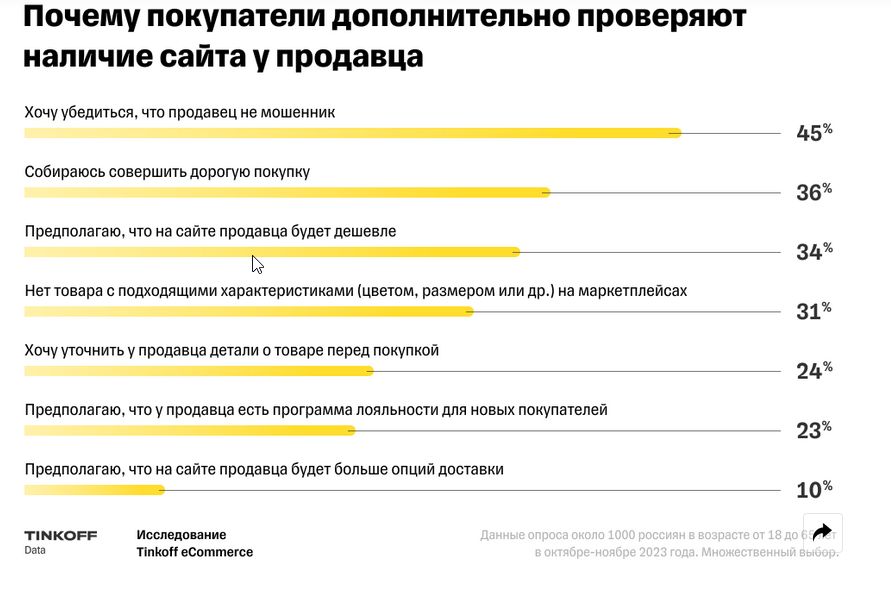
В 2023 году активно изучались и предпочтения россиян. По данным опроса агентства Aquarelle Research и Яндекс. Рекламы, проведённого в прошедшем году и охватившего более более 1,5 тыс. россиян, проживающих в городах РФ с населением от 100 тыс. чел., порядка 25% респондентов приобретают в онлайн-торговле хотя бы одну категорию товаров, 29% покупают товары исключительно в оффлайн-торговле, 46% используют оба канала для покупки товаров, относящихся к одной категории.

Треть опрошенных приобретает товары в интернет-магазинах производителей или ретейлеров. Только порядка 13% респондентов совершают покупки хотя бы одной категории товаров исключительно на маркетплейсах.

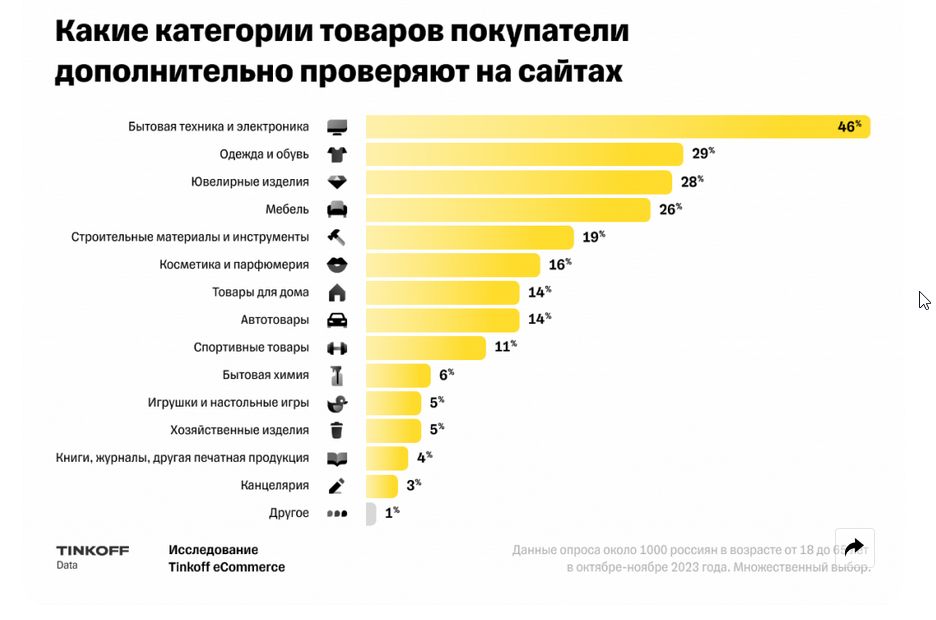
При этом перед покупкой товара 63% от общего числа участников опроса изучают информацию о нём в онлайне. 46% из них используют поисковые системы или геосервисы, 51% – маркетплейсы, 21% – сайты с отзывами. В большинстве категорий товаров основными драйверами покупок выступают цены, условия доставки и предлагаемый ассортимент.

Tinkoff eCommerce в октябре-ноябре 2023 года исследовал, как влияет на доверие покупателей наличие отдельного сайта у продавца. Анализ ответов опрошенных в ходе исследования более 1 тыс. онлайн-покупателей из разных регионов РФ показал, что 75% респондентов для получения дополнительной информации и принятия решения о покупке ищут в интернете сайты магазинов, даже если в итоге приобретут товар на маркетплейсе или другой интернет-площадке.

Почти половина из них (45%) делает это, чтобы убедиться, что продавец – не мошенник. 36% покупателей отметили, что хотят проверить качество приобретаемого дорогого товара перед покупкой, 34% – ищут более выгодные предложения:



Чаще всего перед покупкой пользователи проверяют товары с большим набором характеристик или же с высокой ценой:



Для покупателей важны ряд элементов сайта, влияющих на доверие к продавцу и его товарам, но в первую очередь – полнота информации о товаре и наличие реальных отзывов людей, купивших товар.

67% онлайн-покупателей считают важнейшей характеристикой интернет-продавца его рейтинг и отзывы о нём и предлагаемых им товарах.

**Тренды интернет-торговли на ближайшие годы: ренессанс интернет-магазинов**

Онлайн-торговля будет развиваться в 2025 году как количественно, так и качественно. На прошедшем на прошлой неделе Всемирном экономическом форуму в Давосе большое внимание было уделено «болезни X», смертность от которой якобы в 20 раз выше смертности от коронавируса COVID-19, и возможному наступлению пандемии, связанной с ней. Насколько правдоподобны «страшилки» бюрократов из Всемирной организации здравоохранения, покажет жизнь. Будем надеяться на то, что их прогнозы окажутся «пшиком».

Однако, если новая пандемия действительно свалится на человечество, интернет-торговля получит такой же мощный импульс, как это было в период коронакризиса. И тогда развитие этого сегмента экономики будет столь же стремительным, как в 2020 году, а сценарий Data Insight, о котором рассказывалось выше, реализуется с некоторыми несущественными изменениями во второй раз. Хотя, конечно, лучше было бы, чтобы такой сценарий не реализовывался и всё шло своим чередом — без «болезней X» и прочих напастей.

В 2025 году начнётся ренессанс интернет-магазинов, для которых ключевыми станут такие факторы, как узнаваемость бренда, комплексное продвижение и занимаемые позиции в выдачах поисковиков. Интернет-магазину не просто перейти на новые рельсы, задачу продвижения в интернете своего сайта решить самостоятельно ему сложно. Для этого необходимо привлечение квалифицированных специалистов, прежде всего – интернет-маркетологов.

* + 1. Преимущества интернет-магазинов перед традиционной торговлей.

Все большую популярность в России приобретает интернет-торговля, так как реализация товаров через интернет-магазин является удобной и очень выгодной как для продавцов, так и для покупателей.

Продажа товаров с помощью сети интернет весьма привлекательна для предпринимателей. Онлайн торговля не требует наличия «обычной» торговой площадки, достаточно создать виртуальный магазин, который по сути представляет собой сайт в интернете. Продавцу не нужно приобретать дорогостоящее торговое оборудование, нанимать торговый и обслуживающий персонал.

Плюсы покупок товаров через интернет-магазин есть и у покупателей. Очень часто товары, приобретаемые посредством дистанционной торговли, имеют меньшую цену, чем аналогичные товары в стационарной торговой точке. Покупатель имеет возможность не спеша выбрать необходимую вещь, не выходя из дома, тем более, что компьютерная техника и интернет сегодня есть практически в каждой семье.

При этом торговля через интернет имеет для потребителя некоторые минусы. Чтобы онлайн-покупка не обернулась для потребителя непредвиденными неприятностями, Роспотребнадзор напоминает об особенностях приобретения товаров через интернет.

Недобросовестные предприниматели, осуществляя торговлю через интернет, часто нарушают сроки поставки по оплаченному товару либо не доставляют товар потребителю, реализуют некачественный товар или намеренно вводят потребителя в заблуждение относительно основных потребительских свойств товара. Часто бывает, что при покупке товаров через интернет-магазин трудно обменять или вернуть купленный товар.

Если вы решили приобрести товар дистанционно, в первую очередь обратите внимание, что на странице сайта в интернете обязательно должна быть представлена информация об адресе (месте нахождения) и полном фирменном наименовании продавца, независимо от того является ли организация юридическим лицом или индивидуальным предпринимателем.

Выбирая товар онлайн, покупатель должен видеть всю информацию о товаре, а именно:

* его основные потребительские свойства и адресе (месте нахождения) продавца,
* место изготовления товара, полное фирменное наименование продавца
* цену,
* условия его приобретения и доставки,
* рок службы товара,
* срок годности товара,
* гарантийный срок,
* порядок оплаты товара,
* срок, в течение которого действует предложение о заключении договора.

После того, как покупатель ознакомился со всей информацией о товаре и сделал онлайн-заказ, он вправе отказаться от него в любое время до его передачи, а после передачи — в течение семи дней.

После того, как сделан онлайн-заказ на электронную почту, посредством СМС сообщений, или иным другим способом от продавца должна прийти информация о:

* продавце — полное фирменное наименование и адрес (место нахождения) продавца,
* покупателе — фамилия, имя, отчество покупателя или указанного им лица (получателя),
* адресе, по которому следует доставить товар;
* наименовании товара, артикула, марки, разновидности, количестве предметов, входящих в комплект приобретаемого товара.
* цене товара;
* виде услуги (при предоставлении), времени ее исполнения и стоимости;
* обязательствах покупателя.

Обязательства продавца по передаче товара и иные обязательства, связанные с передачей товара, возникают с момента получения продавцом соответствующего сообщения от покупателя о намерении заказать ту или иную продукцию.

Договор считается заключенным после того как продавец выдал покупателю кассовый или товарный чек, либо иной документ, подтверждающий оплату товара.

Если покупатель осуществляет оплату в безналичной форме или берет предмет торговли в кредит (за исключением оплаты с использованием банковских платежных карт) продавец обязан подтвердить передачу товара путем составления накладной или акта сдачи-приемки товара.

В момент доставки товара, одновременно с ним, продавец обязан представить в письменной форме, путем размещения на продукции, на электронных носителях, прикладываемых к товару, в самом изделии (на электронной плате внутри электротовара в разделе меню), на таре, упаковке, ярлыке, этикетке, в технической документации, всю ту информацию, которая была выложена на сайте.

Для доставки товаров в место, указанное покупателем, продавец может использовать услуги третьих лиц (с обязательным информированием об этом покупателя).

В случае если доставка товара произведена в установленные договором сроки, но по каким-либо причинам покупатель не смог принять товар, последующая доставка производится в новые сроки, согласованные с продавцом, после повторной оплаты покупателем стоимости услуг по доставке товара.

Если по какой-либо причине покупатель решил вернуть покупку надлежащего качества, то продавец возвращает покупателю полную сумму за товар, за исключением расходов покупателя на доставку.

В случае, если информация о товаре, выложенная на сайте не соответствует доставленной продукции или нарушены условия доставки товара, покупатель вправе отказаться от покупки, при этом продавец обязан вернуть покупателю денежные средства.

Если вам доставили товар ненадлежащего качества, вы вправе потребовать:

* безвозмездного устранения недостатков товара или возмещения расходов на их исправление покупателем или третьим лицом;
* соразмерного уменьшения покупной цены;
* замены на товар аналогичной марки (модели, артикула) или на такой же товар другой марки (модели, артикула) с соответствующим перерасчетом покупной цены.

При этом в отношении технически сложных и дорогостоящих товаров эти требования покупателя подлежат удовлетворению в случае обнаружения существенных недостатков.

При возврате товара ненадлежащего качества отсутствие у покупателя документа, подтверждающего факт и условия покупки товара, не лишает его возможности ссылаться на другие доказательства приобретения товара у продавца.

Информация о порядке и сроках возврата товара должна содержать:

* адрес (место нахождения) продавца, по которому осуществляется возврат товара;
* режим работы продавца;
* максимальный срок, в течение которого товар может быть возвращен продавцу,
* предупреждение о необходимости сохранения товарного вида, потребительских свойств товара надлежащего качества до возврата его продавцу, а также документов, подтверждающих заключение договора;
* срок и порядок возврата суммы, уплаченной покупателем за товар.

Даже если вы оплатили товар путем перевода средств на счет третьего лица, указанного продавцом, то все равно это не освобождает продавца от обязанности осуществить возврат уплаченной покупателем суммы при возврате товара как надлежащего, так и ненадлежащего качества.

1.1.3 Основные требования к интернет-магазинам

**Правила онлайна: кто проверяет интернет-магазины**

Объём онлайн-продаж в России за 2024 год может достигнуть 9–11 трлн рублей. Уже в октябре 2024 года рынок преодолел планку в 7 трлн рублей. Какие требования нужно соблюдать, если вы уже начали или только планируете продавать товары через интернет.

Общие правила работы офлайновых и онлайновых магазинов одинаковы: соблюдать право покупателей на качественную продукцию, достоверную информацию, возврат и обмен товаров. Но существуют и специальные требования, которым подчиняются только онлайн-магазины.

Не продавайте запрещённые товары. Интернет-магазины не могут продавать:

* алкоголь;
* табачную продукцию;
* некоторые виды лекарств (нерецептурные медикаменты могут продавать предприниматели и организации с фармацевтической лицензией, а для рецептурных есть отдельный список, закреплённый Приказом Министерства здравоохранения);
* ртутные термометры;
* живых животных и насекомых;
* электрошокеры;
* тесты на определение Covid-19, кроме предназначенных для домашнего использования;
* автомобильные заправленные кислотные и щелочные аккумуляторы;
* неодимовые магниты;
* перекись водорода 30%;
* легковоспламеняющиеся жидкости;
* рентгеновское оборудование, приборы и оборудование с использованием радиоактивных веществ и изотопов;
* медицинское оборудование;
* шифровальную технику (например, устройства для прослушки, перехвата радиоволн);
* оружие (например, охотничьи ружья, травматические пистолеты).

Правила работы с персональными данными.

При первом посещении пользователя сайт собирает куки — первичные данные о том, из какого города и с какого компьютера зашёл пользователь. Затем при заказе просят ввести номер телефона, Ф. И. О., адрес, а при оплате — данные банковской карты.

Чтобы соблюсти требования закона и не получить штраф, разработайте два внутренних документа:

* политику обработки персональных данных;
* согласие на сбор и обработку персональных данных.

Постарайтесь разместить политику на видном месте на сайте, а текст согласия — в форме, в которой покупатель указывает конфиденциальную информацию. Чтобы подготовить документы было проще, воспользуйтесь рекомендациями по составлению политики и образцом согласия на сайте Роскомнадзора.

Указывайте на сайте полную информацию о товаре.

На сайте магазина должна быть полная и достоверная информация о товаре. Вот что нужно указать в карточке каждого товара:

* потребительские свойства товара (например, размер, вес, цвет);
* адрес и полное наименование продавца;
* место изготовление товара;
* цену и условия продажи и доставки товара;
* сроки службы, годности и гарантии;
* порядок оплаты товара.

Если доставленный товар не соответствует описанию на сайте, покупатель может его вернуть в течение 7 дней с момента получения. При этом деньги ему возвратят не позднее 10 дней.

Информацию о компании или ИП тоже нужно разместить на сайте: это требование закона о защите прав потребителей. Одного названия недостаточно, укажите место регистрации, ИНН, ОГРН и режим работы.

Кто приходит с проверкой.

Работу интернет-магазинов контролируют три ведомства: Роскомнадзор, Роспотребнадзор и Федеральная антимонопольная служба (ФАС). Обычно они проверяют сайты по жалобам покупателей, но раз в три года могут включить ваш магазин в план проверок.

Роскомнадзор

Проверяет, как соблюдаются требования о защите персональных данных. Инспекторы Роскомнадзора смотрят, есть ли на сайте политика обработки персональных данных и текст согласия на их получение. Если сотрудники ведомства установят, что интернет-магазин собирал персональные данные, но не спрашивал согласие покупателей, выпишут штраф: до 300 тысяч рублей для ИП и до 700 тысяч рублей — для юрлиц.

Роскомнадзор также проверяет, есть ли на сайте незаконная информация, например призывы к межнациональной розни или экстремизму. Причём инспекторы смотрят не только сами карточки товаров или статьи, но и отзывы к ним. Поэтому не забывайте модерировать комментарии пользователей: перечитывать на предмет этичности и при необходимости удалять.

Роспотребнадзор

Проверяет, как интернет-магазины соблюдают закон «О защите прав потребителей». Вот какие ошибки совершают предприниматели:

* привозят товар, который не соответствует описанию на сайте;
* не указывают на сайте юридическую информацию о компании;
* отказываются оформлять возврат товара;
* доставляют товар с браком или не доставляют вовсе.

Роспотребнадзор запрашивает у интернет-магазина документы по жалобе. Список зависит от проблемы, которую выявили инспекторы или на которую пожаловались клиенты. Например, если она касается доставки, то попросят договор, квитанцию и чек. Если касается состава, свойств или внешнего вида товара, то сертификаты качества.

Антимонопольная служба

ФАС проверяет, как интернет-магазин рекламирует себя и свои товары. Реклама должна быть достоверной и не вводить покупателей в заблуждение. Если вы напишете, что товар стоит 20 тысяч рублей, а в магазине на ценнике написано 25 тысяч рублей, то инспекторы могут выписать вам штраф на сумму до 500 тысяч рублей.

Кроме того, антимонопольная служба следит, чтобы предприниматели боролись за потребителя честными методами. Например, если вы пишете «лучший товар в России», но никак не подтверждаете эту информацию, ФАС посчитает ваши действия актом недобросовестной конкуренции.

* 1. **Обзор технологий для разработки веб-приложений**
     1. Серверные языки программирования (PHP, Python, Node.js)

PHP.

PHP или PHP Hypertext Preprocessor — один из лучших языков для веб-разработки. PHP — это динамический язык программирования общего назначения с открытым исходным кодом, созданный в 1994 году. Он используется для разработки серверных приложений и может встраиваться в HTML. PHP не требует компилятора и может работать практически на всех основных операционных системах, таких как Windows, Linux, macOS и т. д. Он очень популярен среди веб-разработчиков благодаря простоте освоения, кроссплатформенной совместимости, функциям ООП и возможности взаимодействия с системами баз данных, включая Oracle и MySQL.

Это основной язык для веб-разработчиков, поскольку на нём создано более 80% веб-сайтов в интернете. Приложения для социальных сетей Facebook и Yahoo разработаны на PHP.

Первый релиз PHP состоялся в 1994 году. Он представлял собой CGI-скрипт на Perl’е, собирающий статистику о просмотрах резюме разработчика Расмуса Лердорфа. Именно это определило название языка, ведь он вырос из реальной персональной страницы. В то время еще не было понятия Open Source, а все свободные программы распространялись под грифом «freeware» (свободный софт).

По мере развития PHP перестал быть языком для создания персональных страниц, и термин стали применять со значением Hypertext Preprocessor. Система продолжает развиваться – например, в 2020 году официально выпущен 8 релиз PHP. Правда, компания Microsoft, специалисты которой ранее занимались тестированием безопасности, решила отказаться от поддержки языка, но энтузиасты уже взялись искать решение.

Преимущество PHP заключается в том, что это серверный язык. Поэтому он и стал основой для создания веб-приложений – пользователи получают результат вычислений, а всю нагрузку по ним забирает удаленный сервер. Именно это стало причиной появления таких CMS, как WordPress, Joomla, Drupal. На PHP написаны фреймворки типа Laravel, Yii2, Symfony.

Существует три основных направления, где востребован PHP:

* Написание скриптов, исполняемых на стороне сервера. Основное применение PHP – создание сайтов, размещаемых на хостинге (заранее настроенном для этого).
* Написание скриптов, которые исполняются в командной строке независимо от веб-сервера и браузера. Примеры приложений – Task Scheduler под Windows, cron под Linux.
* Разработка приложений GUI, исполняемых на стороне клиента. Основная сфера применения – создание кроссплатформенных приложений.

Язык PHP поддерживается буквально всеми платформами – от Microsoft Windows и macOS до GNU Linux, Unix, RISC OS, OS/2 и других. Также он совместим с распространенными web-серверами – Apache, Personal Web Server, Microsoft Internet Information Server, Netscape, Oreilly Website Pro, iPlanet, OmniHTTPd, Xitami, Caudium.

Перспективы применения PHP.

Сегодня активно применяются аналоги PHP – Angular, React, Node.js, Python. Некоторые разработчики даже довлеют за переход на Java и С. Но «старичок» пока держится и будет держаться долго. Все-таки слишком много сайтов разработано и продолжает разрабатываться с активным использованием функционала PHP (до 79%, если верить статистике W3Tech). Например, одни только сайты на CMS WordPress ежедневно создаются сотнями.

Преимущества использования PHP:

* низкий порог вхождения,
* проверенные инструменты разработки,
* крупное профессиональное сообщество,
* большая библиотека готовых решений,
* нетребовательность к рабочему окружению.

С точки зрения новичка – это хорошая платформа для изучения разработки веб-сайтов. Синтаксис языка входит в группу С-подобных, поэтому после PHP легко осваивать Java, C#, C++ и подобные платформы. Чуть сложнее будет переходить на Python или Goland, но это дело техники и желания освоиться с новыми направлениями.

Платформа продолжает развиваться, в нее внедряются лучшие практики других языков, на ней создаются фреймворки, которые сами по себе являются «новыми языками». PHP только выигрывает от этого благодаря своей надежности и универсальности.

Python.

Питон — это язык программирования, который используется в разных областях. Он не только позволяет создавать веб и мобильные приложения, но и разрабатывать программное обеспечение для ПК. Python — незаменимый инструмент для обработки больших данных, математических вычислений и машинного обучения.

Однако что делает Python настолько привлекательным для программистов? Во-первых, его синтаксис интуитивно понятный, что делает его хорошим выбором для начинающих. Во-вторых, благодаря множеству библиотек и фреймворков, а также своей гибкости, он может быть использован в разных сферах: web-разработке, визуальных интерфейсах, базах данных, сложных расчетах и т.д.

Кроме того, Python применяется в научных исследованиях — используется для анализа данных и создания моделей. Этот язык создает обилие возможностей для науки о данных.

У Python много преимуществ перед другими языками программирования, таких как простота в использовании, мощные библиотеки и фреймворки. В этой части мы рассмотрим основные характеристики Python и узнаем, почему он становится предпочтительным языком для науки о данных.

Python — это ЯП высокого уровня. Python легко выучить по сравнению с другими языками, такими как C++, C#, Javascript, Java и т. д. На языке Python очень легко писать код, и любой может изучить основы Python за несколько часов или дней. Это также удобный для разработчиков язык.

Программы на Python выполняются путем пошагового интерпретирования исходного кода. В отличие от компилируемых языков, таких как C++ или Java, где исходный код сначала компилируется в машинный код, а затем выполняется на целевой платформе, Python выполняется непосредственно из исходного кода.

Интерпретируемость Python имеет несколько достоинств. Во-первых, это делает разработку простой и быстрой. Поскольку нет необходимости вручную компилировать и запускать программу после каждого изменения, разработчики могут немедленно видеть результаты работы.

Во-вторых, интерпретируемость Python обеспечивает переносимость кода. Программы могут быть запущены на разных операционных системах без необходимости изменения исходного кода. Это делает Python удобным выбором для разработки кросс-платформенных приложений и обеспечивает высокую степень переносимости.

Одна из ключевых особенностей Python — объектно-ориентированное программирование. Python поддерживает ООП, концепции классов, инкапсуляцию объектов и т. д. С помощью его проще управлять процессом программирования при создании сложных проектов.

Вы сможете быстро выявить и исправить большинство проблем программы, если поймете, как интерпретировать трассировки ошибок Python. Просто взглянув на код, можно определить, для чего он предназначен. С другими языками программирования это сделать сложнее.

Питон — это высокоуровневый язык программирования. Он прост в использовании. Мощные фреймворки и библиотеки упрощают облегчают программистам работу. У языка дружное комьюнити, где помогают и новичкам, и опытным веб-разработчикам.

Преимущества языка программирования Python являются:

1. Простота и легкость изучения. У Python простой синтаксис, который легко читать и понимать. Это делает его идеальным для начинающих программистов.

2. Библиотеки и фреймворки. Python имеет много библиотек и фреймворков, которые помогают разработчикам ускорить процесс разработки.

3. Кроссплатформенность. Python может работать на разных операционных системах, таких как Windows, Linux, Mac OS и других.

4. Интерактивность. Python позволяет быстро тестировать код благодаря интерактивной консоли.

Однако, как и любой другой язык программирования, у него есть и недостатки, такие как медленная скорость выполнения и несколько ограниченные возможности для разработки низкоуровневых приложений.

Рассмотрим подробнее каждый из них:

1. Низкая скорость выполнения. Python — не самый быстрый язык программирования, поэтому он может быть неподходящим для некоторых приложений, требующих высокой производительности.

2. Ограниченная поддержка многопоточности. У Python ограниченная поддержку многопоточности, что может привести к проблемам с производительностью в некоторых случаях.

3. Недостаточная документация. Некоторые библиотеки и фреймворки Python могут иметь недостаточную документацию, что затрудняет их использование.

4. Ограниченные возможности для разработки мобильных приложений. Python не является лучшим вариантом для разработки мобильных приложений, поскольку он не имеет достаточно мощных инструментов для этой цели.

5. Ограниченная поддержка некоторых областей. Python может быть ограничен в некоторых областях, таких как игровая разработка или разработка приложений для научных вычислений.

Где используется Python?

Python — язык популярный и многие компании использует его в анализе данных, создании веб-приложений и веб-сайтов, анализе финансовых данных и многом другом. Язык применяется в различных сферах, начиная от веб-разработки и вычислений в сфере науки, искусственного интеллекта и машинного обучения, заканчивая системным администрированием, автоматизацией задач и разработкой игр.

Многие популярные веб-сайты, такие как Instagram\*, YouTube и Dropbox, используют Python для своей работы. Благодаря своим мощным библиотекам и фреймворкам для обработки и визуализации данных Python используется в научных исследованиях и анализе данных.

Python — один из самых популярных языков программирования для машинного обучения и искусственного интеллекта. Но не во всех областях его возможно применять, так как он ограничен заложенными в него инструментами и технологиями и об этом расскажем подробнее.

Веб-разработка

Одна из областей, в которой проявляет себя Python — это веб-разработка. Питон предлагает множество фреймворков на выбор, включая Bottle.py, Flask, CherryPy, Pyramid, Django и web2py. Эти платформы использовались для поддержки некоторых самых популярных сайтов в мире, таких как Spotify, Mozilla, Reddit, Washington Post и Yelp.

Учебные пособия и статьи в этом разделе охватывают методы, используемые при разработке веб-приложений Python, и фокусируются на том, как программировать реальные решения проблем, которые на самом деле хотят понять обычные люди.

Автоматизация с помощью скриптов Python

Язык программирования скриптов автоматизирует задачи, которые обычно выполняются людьми. В программировании часто используются скрипты Python для автоматизации различных повседневных задач: переименование множества файлов одновременно, преобразование типа файла, загрузка содержимого, основной анализ журналов и многое другое.

Наука о данных и машинное обучение

Наука о данных и машинное обучение — это две взаимосвязанные области, используемые для анализа больших объемов данных и создания алгоритмов, которые позволяют компьютеру самостоятельно обучаться на основе этих данных. Python – один из наиболее популярных языков программирования, который нужен для работы в этих областях.

У Питона много инструментов, которые упрощают работу с данными и машинным обучением. Например, библиотека NumPy позволяет работать с массивами данных, а библиотека Pandas — с таблицами и базами данных. Библиотека Matplotlib позволяет создавать графики и диаграммы для визуализации данных.

Для машинного обучения в Python используются библиотеки Scikit-learn и TensorFlow. Scikit-learn предоставляет инструменты для создания моделей машинного обучения, а TensorFlow – для создания нейронных сетей. В науке о данных разрабатывают множество алгоритмов и моделей на Python для анализа и обработки больших объемов информации.

Научные исследовани

Научные исследования — это процесс поиска новых знаний и решения сложных проблем в медицине, экономике, физике и других. Сегодня наука о данных и машинное обучение стали неотъемлемой частью научных исследований. И здесь Python играет ключевую роль.

Python применяют в различных областях научных исследований. Например, в медицине Python используется для анализа медицинских данных и создания моделей машинного обучения для диагностики заболеваний. В экономике Python используется для изучения финансовых данных и прогнозирования экономических показателей. В физике Python используется для моделирования сложных систем и анализа экспериментальных данных.

Тестирование

В автоматизированном тестировании можно увидеть использование Python и оно даже допустимо, несмотря на то, что основным языком этой области является Java, начинающая уступать Kotlin. Но некоторые небольшие компании отдают предпочтение Python, например, Яндекс.Танк.

В заключение можно подчеркнуть, что Python — это язык программирования, который остается актуальным и востребованным в науке о данных и машинного обучения. У питона много сфер применения.

Node.js

Node.js – вторая по популярности программная платформа после Python. В основном ее используют для создания веб-серверов, хотя технически платформа позволяет установить пакеты и запустить их практически на любых устройствах. Возможно, популярность Node в том числе обусловлена тем, что для разработки приложений используется язык JavaScript – фактически единственный язык фронтендеров, и последние с помощью Node.js получают возможность «выйти за рамки» браузеров создавать приложения для серверов или изолированных устройств.

Node.js – что это такое

Начнем с истоков – с JavaScript. JS – это язык программирования, который создавался специально для того, чтобы расширить возможности браузеров по отображению информации: переходы, интерактивные кнопки, сменяющиеся картинки и другие красивые вещи на страницах сайтов в основном написаны на JS. Со своей задачей JS прекрасно справлялся, и на рынке он был единственным языком, не требующим для своего изучения пары лет – поэтому довольно быстро JavaScript стал основным (и практически единственным) языком для создания фронтэнда.

Но на этом, собственно, все. Дело в том, что язык программирования – это просто соглашение о том, как писать команды, без движка, исполняющего написанные команды, язык не представляет собой никакой ценности. JavaScript исполнялся движками браузеров, поэтому за пределами браузеров его было сложно как-то применить – скорее всего, какие-то самописные инструменты разработки под JS вне браузеров существовали, но большого применения они не нашли.

Все изменилось, когда Google выпустила новый движок для своего браузера Chrome – V8. У V8 было много отличительных особенностей, но в контексте Node.js нас интересует одно конкретное: движок V8 мог работать отдельно от браузера. JavaScript работает в браузерном движке, браузерный движок V8 можно поставить вне браузера – вопрос создания среды разработки на JavaScript, выходящей за рамки браузера, стал вопросом времени. Так 27 мая 2009 года Node.js и появился – изначально как фреймворк + среда выполнения для серверов, а потом и как среда выполнения для любых устройств.

Как это работает

По сути своей Node.js работает так же, как и JVM (Java Virtual Machine). Вы пишете код, после чего c помощью NPM (менеджера пакетов) отдаете его и все необходимые зависимости движку V8 – который, в свою очередь, переводит код в машинный. Этот машинный код впоследствии может быть запущен на любом устройстве, на котором установлен V8.

В качестве сравнения мы привели JVM, но по факту так работают многие кросс-платформенные языки программирования. Что важно понять: этот механизм позволяет запускать код JavaScript где угодно, потому что за низкоуровневое исполнение кода отвечает не разработчик, а движок V8.

Причины востребованности

Есть 3 основные причины, по которым Node.js так популярен:

JavaScript.

Event-driven модель.

Пакеты

Начнем с языка – JavaScript. JS – довольно легкий язык в плане освоения, кроме того, он используется при разработке фронтэнда. Когда появился Node.js, многие фронтэндеры поняли, что теперь они могут писать код не только для фронтэнда, но для бэкенда – то есть появилась перспектива стать фулстэк-разработчиком, которому существенно больше платят. Поскольку фронтэндеров – много, Node.js уже при появлении обзавелся большой базой разработчиков – вот и первая причина популярности.

Вторая, более техническая часть – event-driven модель работы и «встроенная» асинхронность. Большинство языков программирования рассчитано на синхронную работу кода – строки кода выполняются одна за другой, для того, чтобы «активировать асинхронность», вам нужно сделать запрос с помощью специальных инструментов. Для JavaScript эта схема работала бы очень плохо (ввиду того, что команда запроса данных по сети исполняется в миллионы раз медленнее, чем команда вычисления чего-либо на процессоре), поэтому в язык изначально была положена event-driven модель вместе с асинхронностью: команды исполняются по мере поступления и в таком порядке, чтобы какой-нибудь ожидающий процесс не блокировал весь поток. Эта модель языка, унаследованная в Node.js, оказалась очень выгодной для серверных приложений: им как раз обычно не хватает оптимизированной реакции на события и встроенных инструментов неблокирующего ввода/вывода. Все это в сумме приводит к тому, что серверные приложения на Node.js крайне быстро работают и не требуют при этом больших усилий в разработке.

Наконец, третья причина востребованности – npm. Npm – это утилита, которая позволяет устанавливать пакеты, то есть готовые решения от сторонних разработчиков. Поскольку Node.js – платформа с открытым исходным кодом, большинство готовых решений лежат в открытом доступе и распространяются бесплатно. Кроме того, этих пакетов – крайне много, и иногда можно решить сложную задачу в несколько команд npm и пару строк кода – установить npm, установить пакеты, кодом связать готовые решения.

Минусы Node.js

Node.js – не панацея от всех бед, как и у любой другой технологии, у Ноды есть свои недостатки – вот они:

Однопоточность. Несмотря на асинхронность, Node.js утилизирует только одно ядро процессора – это делается с помощью сложного планировщика задач. Однопоточность не является минусом, пока мы обрабатываем тысячи мелких задач, но если Node.js получит на вход одну крупную задачу – она забьет весь поток, что по «симптомам» очень похоже на блокирование ввода/вывода. Эту проблему можно обойти специальными библиотеками, но с ними нужно уметь работать.

Нестабильные библиотеки, неразвитость некоторых областей. Обе проблемы исходят из того, что пакеты npm в основном разрабатываются «всем миром», то есть нет какой-то централизации – из-за этого разные версии пакета могут существенно отличаться друг от друга, а менее востребованные пакеты просто некому развивать.

Легко написать плохой код. Асинхронность, заложенная в Node.js, требует своих жертв – для того, чтобы ее правильно использовать, нужно использовать колбэки/промисы/async-await. Эти инструменты языка нужно хорошо знать, потому что те же колбэки при неправильном использовании приведут к «callback hell», и поддерживать код станет очень сложно.

Где используется Node.js

99.9% – это серверная сторона приложений. Node.js используется там, где нужна скорость, в том числе – в приложениях с мгновенным обменом информацией и приложениях, в которых нужно работать с десятками тысяч пользователей одновременно.

* + 1. Фреймворки для backend-разработки (Laravel, Django, Express.js)

Laravel (в кратце). Более подробно будет изложено в пункте 1.3 дипломной работы.

Появление фреймворков сделало программирование намного более доступным. После этого разработка ПО перестала быть уделом узкого круга профессионалов. Тем более – многие подобные программные платформы распространяются бесплатно. Типичным примером одного из самых популярных бесплатных фреймворков выступает Laravel.

Django

Django — один из самых популярных фреймворков для Python. Его используют при создании современных веб-приложений.

Python — многофункциональный язык программирования. Его применяют для аналитики, Big Data и создания бэкенда сайтов. Но писать на «голом» языке долго и дорого.

Поэтому для коммерческой разработки используют фреймворки — своеобразные каркасы приложений, состоящие из готовых участков кода. Они позволяют оптимизировать процесс разработки. Программисту не нужно с нуля изобретать велосипед, достаточно ввести пару команд и получить работающий модуль.

Фреймворки пишут на конкретном языке программирования под разные специфические задачи. Django создали конкретно для веб-разработки и сейчас это один из самых популярных инструментов для создания современных сайтов.

Фреймворк начали разрабатывать в 2003 году. Он родился в недрах редакции газеты Lawrence Journal-World. Тогда создание сайтов требовало от разработчиков повторения многих рутинных действий, но со временем стало понятно, что их можно упростить и унифицировать.

Для этого было разработано решение в виде первой версии Django. Имя ему дали в честь гитариста Джанго Рейнхардта.

Первоначально цель фреймворка была простой: создавать и обновлять новостные сайты. Раньше это приходилось делать вручную, что занимало много времени.

Первую версию продукта выпустили в 2005 году. Новый фреймворк имел панель инструментов и удобную систему шаблонов, с помощью которых было легко собрать новостную ленту.

С первых дней своей разработки Django отличался мощностью и гибкостью. Его создатели — Эдриан Головатый и Саймон Виллисон сразу сделали исходный код публичным.

Фреймворк быстро завоевал популярность в профессиональном сообществе, а над его улучшением начали работать программисты по всему миру.

Где используют Django

Django — самый востребованный фреймворк для веб-разработки на Python в 2024 году. На его основе работают масштабные сайты с миллионами пользователей. Секрет успеха — в многогранности.

Django подходит для создания:

* Новостных сайтов и платформ для дистрибуции контента — это было его изначальной функцией;
* CMS — систем управления контентом, на которых работают админки сайтов;
* CRM — систем управления процессами в организации и взаимодействия с клиентами;
* Сервисов для бронирования;
* Корпоративных порталов и платформ для ведения документооборота.

Опционально на этом фреймворке пишут также алгоритмические генераторы и платформы для анализа данных, систем фильтрации и машинного обучения.

За счет хорошей оптимизации он позволяет обрабатывать большие объемы информации и выдерживать интенсивный трафик, поэтому его используют для высоконагруженных платформ.

Из чего состоит Django

Этот фреймворк предлагает разработчику готовый набор инструментов для создания сайтов. Он использует принцип MVT или Model-View-Template, который дает возможность разделить логику работы сайта, его внешнее отображение для пользователей и реакции.

Как это работает:

Шаблоны — они определяют, каким будет внешний вид страницы сайта. Это файлы с HTML-кодом, отображающим содержимое. Они могут быть как статическими, так и динамическими;

Представление — отвечает за данные, а также принимает, обрабатывает и реагирует на HTTP-запросы. За счет него приложение отвечает на действие пользователей;

Модель — это источник информации о данных. Она обеспечивает внутренние функции, например, подключение к базе данных или форматирование информации. Ее задача — отвечать за бизнес-логику, свойства и другие манипуляции.

В состав фреймворка входит много модулей, позволяющих реализовывать функционал сайта. В их числе:

* Модули защиты от всех популярных сетевых атак;
* Веб-сервер, позволяющий проводить разработку и тестирование;
* Интерфейс администратора, через который можно вносить изменения во внутреннее содержимое сайта;
* Встроенные модули авторизации и аутентификации;
* Диспетчер, который обменивается событиями между компонентами;
* Фильтры обработки запросов — перенаправления, кэширования и так далее;
* Модули для подключения внешних блоков к сайту.

Преимущества Django

Фреймворк высоко ценят в профессиональном сообществе за целый ряд объективных преимуществ, которые сделали его лидирующей технологией в веб-разработке.

Полномасштабная экосистема

Django включает в себя все инструменты и модули для веб-разработки. Программисту достаточно только его, чтобы создавать современные высоконагруженные приложения.

Он не требует дополнительных инструментов и замкнут сам на себе. При этом в его экосистему входит множество дополнительных модулей, которые полностью закрывают потребности разработчика.

Фактически из Django можно собрать сайт как из конструктора, используя и модернизируя готовые решения.

Простота

Легкость изучения Django — наследие языка программирования, на котором этот фреймворк и написан. Если разработчик освоил Python, то Django вряд ли вызовет затруднения. Хотя и потребует некоторого времени на изучение.

Универсальность

Django подходит для сайтов любого типа. На нем можно писать видеохостинги, социальные сети, новостные порталы, интернет-магазины и т.д.

К нему подключают любые базы данных, Django работает со всеми типами клиентских сред и файлов. Это делает его универсальным решением.

Большое комьюнити

Django — старый фреймворк, который за время своего существования оброс многочисленными сторонними модулями и дополнительными инструментами. Это позволяет разработчику найти готовое расширения для любой нестандартной задачи.

Также фреймворк хорошо задокументирован — есть возможность найти справочную информацию даже на русском языке. У Django большое и старое сообщество, которое помогает новичкам с поиском ответов.

В комьюнити фреймворка на Github более двух тысяч участников, которые занимаются его развитием и поддержкой.

Безопасность

Django дает возможность проектировать сложные, высоконагруженные, но при этом безопасные приложения. С ним не придется жертвовать скоростью ради сохранности данных.

А все благодаря встроенной системе защиты. Она хорошо работает против скриптов, SQL-инъекций и подделки запросов.

Сообщество уделяет пристальное внимание вопросам безопасности и регулярно выпускает обновления, которые помогают бороться с новыми угрозами.

Объектно-реляционное отображение (ORM)

Встроенный ORM — библиотека, позволяющая настроить взаимодействие с различными базами данных и переносить информацию из БД к объектам. Она помогает подключать одновременно несколько баз.

Эта функциональность дает возможность ускорить разработку и уменьшить нагрузку на программиста. Ему не обязательно знать язык, который использует конкретная база данных.

Также с помощью этой библиотеки можно быстро переключаться между базами. Эта опция не требует большого объема кода.

Подходит для SEO

Django создавали для новостных сайтов, поэтому в нем много внимания уделили вопросам оптимизации и получения поискового трафика. Сайты, написанные на этом фреймворке, хорошо индексируют поисковики.

Для этого в нем есть несколько встроенных инструментов. Например, возможность автоматически создавать семантические URL для страниц.

Встроенная панель администратора

У Django, в отличие от других фреймворков для веб-разработки, есть встроенная панель администратора, которую он самостоятельно генерирует во время создания сайта.

Панель дает возможность быстро и эффективно работать со всеми объектами баз данных. Также ее легко модифицировать под нужды конкретного приложения.

Фреймворк «из коробки» оснащен многими полезными инструментами для создания профилей пользователей, модулей аутентификации и разрешений.

Масштабируемость

Эта характеристика фреймворка определила его популярность в коммерческой разработке. Масштабируемость — сильная черта Django. Она позволяет создавать сайты «на вырост», которые в будущем смогут обрабатывать максимальные объемы информации.

Для этого у фреймворка есть компоненты по умолчанию, которые не связаны с остальной функциональностью. Их можно подключать и отключать по необходимости.

Легкая поддержка

Django предоставляет все возможности для повторного использования кода, что сокращает время на разработку и поддержку приложений. Также он помогает группировать функции в отдельные модули. За счет этого структура сайта становится максимально простой.

Фреймворк делает код более читабельным и простым для понимания стороннего разработчика. Таким образом, при смене команды будет сравнительно несложно вникнуть в архитектуру и обеспечить качественную поддержку.

Поддержка библиотек

Они упрощают процесс разработки и хорошо решают специализированные задачи. Django оставляет программисту возможность использовать библиотеки, которые хранят готовые решения для многих задач.

У этого фреймворка есть отдельные библиотеки для управления контентом, работы со сторонними API, создания учетных записей, форм авторизации и регистрации.

Недостатки Django

Даже у эффективных технологий есть свои минусы:

Избыточность

У Django есть все инструменты для создания высоконагруженных приложений. А вот для небольших сайтов их, зачастую, слишком много. За счет этого, не всегда есть смысл применять фреймворк для таких проектов. Тем более, что у него есть более простые альтернативы.

Функции ORM

На сегодня этот компонент Django устарел и не дотягивает до современных стандартов. Ключевой минус ORM — отсутствие поддержки SQLAlchemy. Сейчас это основной инструмент для работы с базами данных у Python.

Отсутствие поддержки WebSocket

Этот протокол позволяет браузеру и серверу обмениваться информацией в режиме реального времени, но фреймворк не поддерживает данную функцию. Однако ее можно реализовать через сторонние решения.

Большой объем информации

Django — единый инструмент и полноценное коробочное решение для создания веб-приложений. Минусы такого подхода проявляются в необходимости осваивать сразу весь объем знаний по фреймворку, даже если все инструменты в данный момент не нужны. А разработчику при создании сайта нужно приложить больше усилий для внедрения сторонних инструментов.

Express.js

Express – фреймворк, написанный на языке JavaScript, который применяют в разработке бизнес-логики мобильных приложений и сайтов. Express работает в Node.js – среде выполнения кода для языка JavaScript. Express относится к микрофреймворкам, отличается гибкостью кастомизации и не диктует шаблонов проектирования бизнес-логики. Тем не менее, разработка на Express чаще подразумевает использование архитектуры MVC (Model-View-Controller) – Модель-Представление-Контроллер.

Поскольку пользовательская часть сайтов работает исключительно на JavaScript, использование Express в качестве фреймворка позволяет задействовать JavaScript в пользовательской и серверной части веб-приложения. Благодаря фулстек-разработке на JavaScript, можно переиспользовать часть кода во фронтенде и бекэнде.

Express является одним из первых бекэнд-фреймворков на JavaScript. По данным BuiltWith, Express используется в разработке в разы больше, чем его основные конкуренты – бекэнд-фреймворки Meteor и Sails.js.

Express является частью популярного стека программирования MEAN. Технологии, которые входят в стек MEAN:

* M – NoSQL-база данных Mongo.DB;
* E – бэкенд-фреймворк Express;
* A – фронтенд-фреймворк Angular;
* N – среда выполнения JavaScript-кода Node.js.

Также Express используется в стеках MERN и MEVN, где в качестве фронтенд-фреймворков выступают React и Vue.js.

Где применяют Express

Чаще всего Express применяют в разработке бизнес-логики сайтов, реже – в разработке мобильных и десктопных приложений.

Преимущества разработки на Express

* популярность фреймворка среди русскоязычного сегмента;
* возможность разработки как маленьких, так и больших проектов;
* переиспользование одного и того же кода на JavaScript в пользовательской и серверной части приложения;
* скорость работы приложения за счет использования технологии неблокирующего ввода/вывода в среде Node.js;
* фреймворк не навязывает архитектуру бизнес-логики приложения, обеспечивая более широкие возможности кастомизации;
* регулярные обновления фреймворка.

Недостатки разработки на Express

* из-за свободы в работе с фреймворком, новые разработчики могут дольше погружаться в действующий проект с неочевидной архитектурой;
* разработчики Express усиленно продвигают Koa.js, который является обновленной версией Express.
  + 1. Frontend-технологии (HTML/CSS, JavaScript, Vue.js/React)

HTML/CSS.

HTML — это язык гипертекстовой разметки, который отвечает за структуру веб-страницы. Другими словами, это основа, без которой сайт не мог бы существовать. HTML-разметка создает структуру сайта: заголовки, абзацы, списки, ссылки, изображения и таблицы. А HTML теги и атрибуты помогают браузеру понять, как нужно отобразить контент.

HTML-документ состоит из различных элементов, каждый из которых выполняет свою функцию:

* Заголовки (<h1> – <h6>) используются для обозначения заголовков разного уровня. <h1> самый важный, <h6> наименее значимый.
* Абзацы (<p>) предназначены для форматирования текстовых блоков.
* Списки (<ul>, <ol>, <li>) применяются для создания маркированных и нумерованных списков.
* Ссылки (<a href="URL">) позволяют переходить на другие страницы или ресурсы.
* Изображения (<img src="URL" alt="Описание">) вставляют графику на страницу.
* Таблицы (<table>, <tr>, <td>) используются для представления данных в табличной форме.

CSS — это язык стилей для оформления веб-страниц, созданных с помощью HTML. Он отвечает за визуальное представление: цвета, шрифты, отступы, расположение элементов и даже анимацию. Когда браузер загружает HTML-страницу, он сначала считывает ее структуру, а затем применяет CSS-стили, чтобы придать элементам нужный вид. Например, с помощью CSS можно задать:

* Цвет текста и фона.
* Отступы между элементами.
* Размеры блоков и изображений.
* Шрифты и их параметры.
* Анимации и эффекты при наведении курсора.

HTML и CSS применяются в большинстве проектов, связанных с веб-разработкой. Эти технологии лежат в основе всего, что существует в интернете — от простых блогов до сложных веб-приложений. Разработчики могут использовать их для решения следующих задач:

* Создание веб-сайтов. HTML отвечает за структуру страницы, а CSS — за ее оформление. Независимо от того, пишете ли вы личный блог, интернет-магазин или корпоративный сайт, без HTML и CSS не обойтись. Даже крупные платформы, такие как YouTube и Wikipedia, используют эти технологии.
* Работа с емейл-рассылками. Многие рассылки оформляются с помощью HTML и CSS. В почтовых сервисах, таких как Gmail и Outlook, используется специальная разметка для красивого отображения писем.
* Создание веб-интерфейсов и панелей управления. Большинство онлайн-сервисов, например интернет-банкинг, CRM-системы или административные панели сайтов, разрабатываются с применением HTML и CSS.
* Разработка мобильных приложений. Гибридные мобильные приложения (созданные с помощью технологий React Native или Apache Cordova) используют HTML и CSS для формирования интерфейса, который адаптируется под разные устройства.
* Работа с системой управления контентом (CMS). Платформы, такие как WordPress, Joomla и Drupal, используют HTML и CSS для создания и кастомизации шаблонов сайтов. Понимание этих технологий позволяет гибко управлять дизайном.

JavaScript

Инструмент JavaScript (сокращенно JS) относится к языкам программирования высокого уровня с возможностью встраивания в другие приложения. Все типы функциональных модулей создаются в виде сценариев. По синтаксису он схож с языком Java, но общая у них только часть наименования. Платформа регулярно обновляется, код становится все более рациональным.

Особенности языка:

* JS изначально разрабатывался по принципу «пиши меньше – делай больше».
* Код поддерживает функциональные, императивные и событийно-ориентированные стили.
* Платформа распространяется бесплатно без лицензионных отчислений разработчикам.

Синтаксис JavaScript подходит не под все задачи. Это привело к появлению ряда искусственных языков, которые транспилируются (конвертируются) после запуска в браузере. Разработчики пишут код на понятном им языке, а платформа «на лету» преобразует его в JS. Примеры: CoffeeScript, Flow, Dart, TypeScript, Brython.

Овладеть основами JavaScript полезно всем, кто касается сферы разработки и продвижения сайтов. В составе любого ресурса есть хотя бы 3-4 скрипта – от счетчика Яндекс.Метрики до формы захвата контактов, виджетов социальных сетей или регистрации аккаунта. Популярность платформы имеет объяснение: язык безопасен, он не предоставляет низкоуровневый доступ к процессам сервера.

Скрипты JS используются в следующих направлениях:

* клиентская часть любых веб-приложений,
* интерактивные элементы интерфейсов на AJAX,
* механизм выдачи Push-уведомлений по модели Comet,
* программы, совместимые с Android, iOS, Windows Mobile,
* браузерные операционные системы типа WebOS,
* макросы для автоматизации рутинных офисных операций,
* приложения, запускаемые на серверах C, C++, Java, Go.

Отдельно стоит упомянуть пользовательские скрипты в браузерах. Ими реализуются такие фишки, как автоматическое заполнение форм, форматирование контента на странице, скрытие и отражение содержимого в зависимости от региона. Аналогичным образом пишутся расширения, плагины (виджеты) для популярных CMS вроде WordPress, MODX, 1С-Битрикс.

Использование скриптов JavaScript обусловлено желанием добавить функции, которых нет и не будет в HTML/CSS. Например, обработку ввода контактов (отправка на указанный email, выдача сообщений, автоматический обратный звонок). За счет мультипарадигменности язык JS способен выполнять разнотипные операции.

* Математическое вычисления. Скрипты заменяют калькуляторы для расчета стоимости и объема материалов, по результату выдаются определенные сообщения. Например, если на сайте представлены тесты, JS отображает описание, соответствующее набранным баллам.
* Обработка и валидация данных в HTML-формах. Вариантов использования множество – от определения, все ли поля заполнены, до проверки соответствия указанному формату (цифры, буквы, их сочетание). Такие скрипты способны работать без обращения к серверу.
* Взаимодействие с пользователем. Сюда относятся выпадающие меню, всплывающие окна и иные эффекты, воспроизводимые при появлении заданных условий. Например, когда был клик по «плюсику» или когда курсор вышел за пределы окна.
* Взаимодействие с HTML-элементами на странице. Типичный пример такой реализации – это конструктор лендингов. Мышью перетаскиваются блоки, по клику на левую/правую кнопку меняется шрифт, цвета и другие параметры.
* Графические эффекты, анимация. Сюда можно отнести передвигающиеся объекты вроде снежинок или таймера обратного отсчета. Также речь идет о плавной прокрутке страницы, появлении и сокрытии объектов, например, при активации слайдера в режиме «карусель».

Все перечисленные типы операций легко обнаружить на одной посадочной странице. Это указывает на востребованность языка JavaScript. Тем более на рынке сайтостроения все больше утверждаются разнообразные конструкторы – от визуальных редакторов для CMS до самостоятельных платформ, где никак не обойтись без скриптов JS.

* + 1. Системы управления базами данных (MySQL, PostgreSQL)

MySQL

На текущий момент, одной из известнейших систем управления базами данных в Интернете является MySQL. Это свободная СУБД, широко используемая такими крупными игроками на мировом IT-рынке, как Facebook, Twitter, YouTube и Airbnb. MySQL предлагает надежность, хорошую производительность и возможность масштабирования, что делает ее одним из лучших выборов для хранения и обработки данных в Интернете. Согласно информации сайта DB-Engines, MySQL находится на 2-ой позиции в рейтинге популярности СУБД. По данным на июль 2023 года им пользуются 18,56% из числа опрошенных респондентов.

Что такое MySQL?

MySQL - это система управления базами данных (СУБД), позволяющая хранить, организовывать большие объемы данных, и манипулировать ими.

MySQL разрабатывается и сопровождается Oracle, и доступна в двух вариантах: коммерческой (MySQL Enterprise Edition) и бесплатной (MySQL Community Edition). Версия Community наиболее распространена.

MySQL предоставляет мощные функции обработки данных. Этот программный продукт использует стандартный язык SQL для обработки данных, При этом, рассматриваемый продукт дает пользователю возможности для создания таблиц, выполнения запросов, вставки, обновления и удаления данных без знания языка запросов SQL.

MySQL сочетает в себе хорошую производительность, высокую надежность и простоту, что делает рассматриваемую СУБД популярным выбором у создателей и администраторов веб-сайтов, требующих обработки данных.

Вот некоторые весьма распространенные задачи, под которые хорошо "заточен" MySQL:

* Хранение данных: MySQL позволяет сохранять большие объемы данных в структурированном виде. Решает задачи надежного хранение данных и дает возможность быстро получать доступ к ним.
* Управление данными: MySQL дает отличные возможности для управления данными, например, создание, изменение и удаление таблиц, индексов и отношений между ними. Дает контролировать и организовывать данные в удобном формате.
* Обеспечение высокой производительности.
* Масштабируемость: в MySQL заложена возможность горизонтального и вертикального масштабирования.
* Гибкость и расширяемость: MySQL содержит широкий набор функций. Есть возможность их настраивать и расширять в соответствии с изменяемыми требованиями.
* Организация и поддержка безопасности данных.

В общем, MySQL является отличным программным средством для работы с данными, которое применимо в областях, где нужна эффективная во всех смыслах система управления базами данных.

Преимущества MySQL

* Плюсы MySQL перед иными СУБД вытекают из задач, под которые данный программный продукт «заточен». Рассмотрим основные из них.
* Уже упоминавшаяся безопасность: MySQL имеет в своем функционале различные функции безопасности, включая возможность установки привилегий пользователя, шифрование данных, аутентификацию и аудит, чтобы обезопасить информацию, хранящуюся в базах данных.
* Производительность: MySQL обладает хорошей производительностью и быстродействием благодаря оптимизированным алгоритмам выполнения запросов. Умеет работать с большими объемами данных с минимальными задержками.
* Масштабируемость: MySQL может использоваться как для небольших веб-приложений, так и для серьезных корпоративных систем. СУБД предлагает различные методы масштабирования, включая горизонтальное и вертикальное масштабирование, что дает возможность расширять базы данных при возникновении необходимости. Это позволяет обрабатывать большое количество одновременных запросов и поддерживать беспрепятственный доступ к данным для пользователей.
* Гибкость: в MySQL реализованы различные типы данных, индексы, хранимые процедуры, триггеры и другие функции, обеспечивающие гибкость при обработке данных. Это дает возможность создания сложных схем данных. MySQL совместима с довольно большим количеством операционных системам.
* Бесплатность и открытые исходники: рассматриваемый продукт имеет открытый исходный код и может быть использован бесплатно. Это делает его доступным для большого круга разработчиков и пользователей.
* Общность и поддержка: MySQL имеет большое сообщество разработчиков и пользователей, что дает возможность доступа к различным ресурсам и помощи при возникновении проблем.

СУБД MySQL предлагает множество преимуществ, делающих ее популярной и широко используемым выбором для работы с базами данных.

Недостатки MySQL

Ниже приведены некоторые недостатки MySQL, на которые стоит обратить внимание:

* Ограничения на производительность: MySQL может иметь проблемы с производительностью при обработке больших объемов данных или когда требуется обработка сложных запросов или подключение большого количества клиентов. Это может привести к замедлению работы или даже отказу в обслуживании.
* Ограниченный набор типов данных: MySQL имеет ограниченный набор типов данных по сравнению с некоторыми другими СУБД. Например, поддержка временных данных и географических данных ограничена по сравнению с некоторыми другими СУБД.
* Ограничения на масштабируемость и доступность: MySQL может столкнуться с ограничениями в масштабируемости и доступности при взаимодействии с большим количеством клиентов. Тогда отказоустойчивость и возможность горизонтального масштабирования могут быть ограничены.
* Сложность администрирования: MySQL может быть сложным для администрирования, особенно для новичков. Настройка и оптимизация параметров конфигурации может быть сложной задачей, требующей глубоких знаний и опыта.
* Безопасность: MySQL может иметь уязвимости в сфере безопасности, например, возможность атаки SQL-инъекцией или недостаточная защита данных. Это требует дополнительных мер безопасности для уверенной надежности и защиты баз данных.

Области применения MySQL

MySQL используется во множестве различных областей и задач. Вот самые значимые сферы применения этой популярной реляционной СУБД:

* Веб-разработка: MySQL часто применяется для построения динамических сайтов и интернет-приложений. При этом СУБД используется в сочетании с веб-серверами (Apache, Nginx).
* Бизнес-приложения: многие предприятия организации используют MySQL для создания и управления баз(ами) данных своих бизнес-приложений. Данная СУБД дает возможность надежно хранить и легко получать доступ к данным, что позволяет эффективно работать с клиентами, заказами, продуктами и другими составляющими бизнеса, получая информацию о них из корпоративных баз данных.
* Аналитика и отчетность: базы MySQL используются как хранилище данных для аналитических и отчетных систем. С помощью рассматриваемого программного продукта можно хранить большие объемы данных, а также выполнять сложные запросы и агрегировать данные, чтобы проводить анализ и создавать отчеты.
* Мобильные приложения: MySQL используется также как СУБД для мобильных приложений, хранящих различные типы данных, такие как пользовательские профили, настройки, результаты и др. Во многом из-за своей надежности и хорошей производительности MySQL хорошо подходит для решения данных задач.
* Встроенные системы: MySQL может быть встроен в другие приложения и устройства для взаимодействия с небольшими базами данных. Это могут быть системы управления контентом (CMS), блоги, форумы, физические устройства, такие как маршрутизаторы и т. д.

Это только основные из сфер применения MySQL. Из-за своей гибкости, надежности и доступности, он массово используется в различных областях, где требуется хранение и управление данными.

СУБД MySQL от компании Oracle представляет собой высокоэффективный и надежный программный продукт для хранения и обработки данных, нашедший свое применение в разнообразных областях и сферах человеческой деятельности.

MySQL имеет множество преимуществ, которые и делают эту СУБД одной из самых популярных в мире.

PostgreSQL

PostgreSQL — это объектно-реляционная система управления базами данных. Она написана на языке C и распространяется свободно. СУБД PostgreSQL работает со стандартным языком запросов SQL, а также с его дополненной версией — PL/pgSQL.

Изначально СУБД носила название Postgres и с самого начала была некоммерческой. Она разрабатывалась в Калифорнийском университете в Беркли с 1986 года по 1994 год. За это время в Постгрес были добавлены правила, процедуры, пользовательские типы и многие другие компоненты, которые считаются основными.

В 1995 году началась разработка новой версии Постгрес, которая получила имя Postgres95. В этом варианте СУБД произошло глобальное изменение: первичный язык запросов POSTQUEL был заменен на популярный SQL.

Со временем Postgres95 была доработана до привычной PostgreSQL. Разработка этой версии вышла за пределы Калифорнийского университета в Беркли и была передана в руки программистов сообщества.

Часто кроме названия PostgreSQL можно встретить вариант Postgres или Постгрес. Несмотря на то, что так называлась первоначальная версия СУБД, это имя также используется для неформального обозначения PostgreSQL.

Любая СУБД имеет уникальные особенности, и с не исключение. Какими преимуществами она обладает:

* Масштабируемость. PostgreSQL позволяет распределить входящие запросы между несколькими инстансами — экземплярами (репликами) БД. Также зачастую на облачных решениях с предустановленным Постгрес можно в любой момент добавить дополнительные ресурсы.
* Кроссплатформенность. СУБД Постгрес можно установить практически на любую известную операционную систему. В число поддерживаемых ОС входят дистрибутивы Linux (Ubuntu, Debian, CentOS и другие), macOS, Windows 10 и 11.
* Безопасность. PostgreSQL позволяет шифровать данные на разных уровнях, а также выбирать средства защиты информации от несанкционированного доступа. В состав Постгрес входит большое количество инструментов безопасности: например, LDAP, SSPI, Kerberos, GSSAPI и многие другие. Эти технологии можно использовать для создания дополнительных опций безопасности а также регулировать настройки доступа к объектам базы данных на разных уровнях: от самой БД до столбцов в ней.
* Поддержка различных типов данных. PostgreSQL может работать не только со стандартными типами данных, которые существуют в большинстве популярных языков, но и со специфическими типами. Среди них можно встретить геометрические расчеты, сетевые адреса, типы полнотекстового поиска и многие другие.
* Расширяемость. Вы можете написать собственные функции для Постгрес на Python, PHP, Java и Ruby. Также для загрузки доступны готовые модули на языке C из репозитория PGXN.
* Открытый код. Постгрес распространяется по лицензии BSD. Это значит, что проект можно использовать, редактировать и распространять среди других пользователей бесплатно. Также разработчики могут модифицировать оригинальный код, скачав его с официального сайта.
* Активное сообщество. Пользователи и разработчики собственного ПО еженедельно получают новости о Postgres из рассылок. Также участники сообщества делятся опытом, ведут дискуссии и помогают в решении вопросов разработки.
* Поддержка NoSQL. PostgreSQL поддерживает форматы XML, JSON и JSONB. Это позволяет записать JSON-документ в базу данных, не разбирая его — от этого производительность сервера БД становится выше.

Также PostgreSQL имеет два недостатка:

* Требовательность к ресурсам. Так как PostgreSQL способна работать со сложными запросами и большими объемами данных, она может требовать больше ресурсов в сравнении с другими популярными СУБД. В моменты высокой нагрузки особенно активно расходуется оперативная память (ресурсы ОЗУ) и процессорное время.
* Сложная настройка. Так как СУБД обладает гибким функционалом для настройки, для работы с ней потребуется знание архитектуры и понимание параметров. Кроме того, необходимо освоить основы языка SQL. Все это может создать сложности в PostgreSQL для начинающих пользователей.
  1. **Особенности фреймворка Laravel**
     1. История и преимущества Laravel

Laravel – это бесплатный PHP-фреймворк с открытым исходным кодом. Laravel разработали в качестве помощника при создании сложных веб-ресурсов и приложений. С его помощью специалисты упрощают процесс аутентификации, а также работу с БД, кэширование, сессии, структуру приложения, маршрутизацию и другие не менее важные процессы.

Возможностей у платформы Laravel немало. Одна из них – построение логичной архитектуры для проектов любой сложности и типа. Платформа характеризуется:

* высокой производительностью;
* возможностью интеграции с другими платформами, а также библиотеками;
* немалым количеством интересных возможностей для разработчиков сайтов и приложений.

Преимущества Ларавел

Универсальность

Нет ограничений в возможностях, доступны любые внедрения сложных сервисов, функций. Проект на Ларавел легко масштабируется. Настроить же админку можно с учетом пожеланий заказчика.

Безопасность

Владелец, да и пользователь, должны быть уверены в безопасности ресурса и веб-приложения. Именно поэтому выбирать технологию следует ту, которая даст гарантии надежности. Laravel framework не допускает возможности взлома, кибератак , также он защищает от скрытых SQL-внедрений в ресурс. Это защищает пользователей от потери важных данных.

Высокая производительность

С помощью Ларавел можно создавать высокопроизводительные приложения. Эта веская причина, чтобы обратить внимание на эту технологию. Также есть возможность интегрировать ее с Memcached и Redis,. Это дает большую гибкость для объектов, которые необходимо кэшировать. Управление же кешированием позволяет сократить время загрузки веб-страниц, что положительно сказывается на ранжировании.

Богатое сообщество

Технология Ларавел имеет открытый исходный код, что привлекает разработчиков, ведь это позволяют создавать приложения без приобретения лицензии. Соответственно, мощное сообщество — огромный плюс, особенно, когда приходится решать сложные задачи.

Laravel шаблоны

Готовые модули — это мощная и замечательная особенность этой технологии. Каждый шаблон Ларавел укомплектован готовыми к использованию плагинами, набором пользовательского интерфейса, списком функций и прочими полезными возможностями, с помощью которых существенно ускоряется процесс разработки приложений. Также шаблонизация не ограничивает специалистов в возможности написать свой код.

Миграция баз данных

Laravel миграции баз данных упрощают создание и управление БД. Также это позволяет сделать совместную работу разработчиков над проектом простой и структурированной. Например, при в любой момент можно откатить изменения. Это минимизирует риск поломки БД.

Архитектура Laravel

Архитектура Laravel — MVC или «Модель-Представление-Контроллер». Такой подход повышает эффективность фреймворка в 10 раз. Несмотря на то, что он используется во многих CMS системах и фреймворках, тут бы мы ее назвали — полная MVС.

Laravel библиотеки

Технология предлагает объектно-ориентированные библиотеки. К примеру, аутентификация с многочисленными функциями, дополнительные модули, позволяющие разработчикам создавать более отзывчивые и структурированные продукты, свободное планирование административной части с помощью готовых решений, у которых есть возможность кастомизации либо написать что то свое. MVC структура проекта позволяет легко внедрять любые библиотеки/модули для работы с сайтом: от простых выводов новостей до систем оплат/оформлений заказа. Также предоставляет фреймворк и дополнительные модули, часть из которых можно приравнять к готовым приложениям. Это позволяет экономить средства заказчика при создании проекта.

Простое тестирование

С помощью встроенной системы тестов разработчики могут выявить проблемы на любом этапе разработки, обеспечить лучший пользовательский опыт. Поскольку Ларавел — передовая технология, то и предлагает она модульное тестирование, которое делает отладку и проверку модуля и его компонентов проще. Соответственно, экономичнее.

К тому же, модульный подход снижает количество ошибок, а значит, приложение с большей вероятностью будет работать без сбоев.

Конкурентоспособные приложени

Любой бизнес, проект должен развиваться. Если еще недавно важно было наличие самого сайта с базовым функционалом, то сейчас пользователям интересны проекты нестандартные, интерактивные. Создать такие не всегда просто, но вот Ларавел позволяет выйти на новый уровень сервиса. С его помощью разработчики могут создавать функциональные и привлекательные приложения, которые позволят вам выделиться среди конкурентов.

Laravel мультиязычность

Локализация Laravel, не просто поддержка многоязычности, а очень удобная система по сравнению с другими фреймворками, которая позволяет быстро и качественно решить эту задачу, помогает выводить бизнес на международный уровень, привлечь новых пользователей. Соответственно, увеличить прибыль.

Скорость разработки

От того, сколько времени потрачено на разработку приложения, зависит и то, сколько придется платить заказчику. Ларавел же позволяет создать высокопроизводительное, функциональное приложение значительно быстрее, чем при использовании других технологий, например, Symfony.

* + 1. Архитектура MVC в Laravel

MVC (Model-View-Controller) – это одна из самых популярных архитектурных моделей, используемых веб-приложениями. Laravel, популярный PHP-фреймворк, также основан на архитектуре MVC. Понимание того, как работает MVC в Laravel, является ключом к разработке высококачественных и легко поддерживаемых приложений.

Архитектура MVC разделяет приложение на три основных компонента:

* Модель (Model) представляет данные, бизнес-логику и правила взаимодействия с базой данных. В Laravel модель отвечает за взаимодействие с таблицами базы данных и содержит методы для извлечения, обновления и удаления данных.
* Представление (View) отображает данные пользователю и предоставляет пользовательский интерфейс для взаимодействия с приложением. В Laravel представления создаются с использованием Blade-шаблонов, которые позволяют вставлять переменные, циклы и условия в HTML-код.
* Контроллер (Controller) обрабатывает входящие запросы от пользователя и взаимодействует с моделью и представлением. Он содержит методы, которые принимают запросы и возвращают ответы. В Laravel контроллеры обрабатывают маршруты и взаимодействуют с моделями для получения или сохранения данных.

Когда пользователь делает запрос веб-приложению на Laravel, маршруты определяют, какой контроллер должен обработать запрос. Контроллер получает данные из модели, обрабатывает их и передает в представление, которое затем отображает данные пользователю.

Использование архитектуры MVC в Laravel позволяет разделить ответственность на различные компоненты и облегчает сопровождение и масштабирование приложения. Разработчики могут легко изменять модели, представления и контроллеры, не затрагивая другие части приложения.

Что такое архитектура MVC?

Модель (Model) отвечает за хранение и обработку данных. Она представляет собой бизнес-логику приложения и взаимодействует с базой данных или другими источниками данных. Модель обрабатывает запросы, проводит валидацию данных и возвращает результаты в контроллер.

Представление (View) отображает данные пользователю и отвечает за визуальное представление информации. Оно может быть в виде HTML-шаблона, графического элемента или другого пользовательского интерфейса. Представление получает данные от контроллера и отображает их в удобной форме для пользователя.

Контроллер (Controller) обрабатывает пользовательские запросы, взаимодействует с моделью и определяет, какие данные должны быть переданы в представление. Он является посредником между пользовательским интерфейсом (представлением) и бизнес-логикой (моделью). Контроллер также может обрабатывать валидацию данных и выполнение других операций, связанных с логикой приложения.

Архитектура MVC помогает сделать приложение более структурированным, упрощает изменение и доработку кода, улучшает его читаемость и позволяет более эффективно работать в команде разработчиков.

Понятие

Модель: модель представляет бизнес-логику приложения и управляет данными. Она отвечает за доступ к базе данных, обработку данных, валидацию, а также реализацию бизнес-правил. Модель предоставляет интерфейс для получения данных и взаимодействия с ними.

Представление: представление отвечает за отображение данных пользователю. Оно использует данные, полученные из модели, и генерирует HTML-код или другой контент для отображения на странице. Представление не должно содержать логики, оно только отображает данные, полученные от контроллера.

Контроллер: контроллер обрабатывает запросы от пользователей и управляет взаимодействием между моделью и представлением. Он принимает запросы от пользователей, вызывает соответствующие методы модели для получения данных и передает их представлению. Контроллер также отвечает за обработку данных, полученных от пользователей, и редиректы.

Архитектура MVC в Laravel обеспечивает разделение ответственности между компонентами и способствует повышению переиспользуемости кода, улучшению тестируемости и облегчению сопровождения приложения. Она позволяет разработчикам легче масштабировать приложение и изменять его, сохраняя при этом чистоту и организованность кода.

Что означает MVC?

Модель (Model) представляет данные и основные бизнес-логику приложения. Она отвечает за взаимодействие с базой данных и обработку данных. Модель также определяет правила валидации данных.

Представление (View) отвечает за отображение данных пользователю. Это место, где пользователь видит результаты работы приложения. Представление получает информацию от модели и форматирует ее для отображения на экране. Оно также обрабатывает пользовательский ввод и отправляет его в контроллер.

Контроллер (Controller) управляет потоком данных между моделью и представлением. Он принимает пользовательский ввод от представления, определяет, какую модель использовать и какие представления отобразить. Контроллер также обрабатывает логику приложения и принимает решение о последующих шагах.

MVC в Laravel позволяет разработчикам эффективно структурировать свои приложения, упрощая поддержку и расширение кода. Он помогает разделить ответственность между разными компонентами и обеспечивает более гибкую и масштабируемую архитектуру.

Что означает MVC?

Модель (Model) представляет данные и основные бизнес-логику приложения. Она отвечает за взаимодействие с базой данных и обработку данных. Модель также определяет правила валидации данных.

Представление (View) отвечает за отображение данных пользователю. Это место, где пользователь видит результаты работы приложения. Представление получает информацию от модели и форматирует ее для отображения на экране. Оно также обрабатывает пользовательский ввод и отправляет его в контроллер.

Контроллер (Controller) управляет потоком данных между моделью и представлением. Он принимает пользовательский ввод от представления, определяет, какую модель использовать и какие представления отобразить. Контроллер также обрабатывает логику приложения и принимает решение о последующих шагах.

MVC в Laravel позволяет разработчикам эффективно структурировать свои приложения, упрощая поддержку и расширение кода. Он помогает разделить ответственность между разными компонентами и обеспечивает более гибкую и масштабируемую архитектуру.

Таким образом, архитектура MVC позволяет разделить логику приложения на три отдельных компонента, что облегчает поддержку, расширение и тестирование кода. Кроме того, MVC позволяет создавать более гибкие и масштабируемые приложения. В Laravel эта архитектура является основой для разработки веб-приложений и помогает разработчикам организовать и управлять сложным проектом.

Принципы работы

Архитектура MVC в Laravel основана на трех основных компонентах: модели (Model), представления (View) и контроллерах (Controller). Каждый из этих компонентов выполняет определенные задачи, чтобы обеспечить эффективное разделение ответственностей и управление работой приложения.

Модель (Model) — представляет бизнес-логику и данные приложения. Она отвечает за взаимодействие с базой данных, выполнение запросов и обработку данных.

Представление (View) — отображает данные и интерфейс пользовательского интерфейса приложения. Он отвечает за отображение данных, управление их форматированием и предоставление пользовательского опыта.

Контроллер (Controller) — управляет потоком данных между моделью и представлением. Он принимает запрос от пользователя, обрабатывает его, взаимодействует с моделью для получения необходимых данных и отправляет эти данные в представление для отображения.

Принцип работы архитектуры MVC в Laravel следующий:

Пользователь отправляет запрос к приложению, например, путем ввода URL-адреса в браузере.

Запрос перенаправляется в маршрутизатор (Router), который определяет, какой контроллер должен обрабатывать запрос.

Контроллер получает запрос, обрабатывает его и взаимодействует с моделью, чтобы получить необходимые данные.

После получения данных контроллер отправляет их в представление, которое отображает данные пользователю.

Пользователь видит результат на своем устройстве в виде отображения данных или выполнения определенных действий.

Такой подход позволяет разработчикам эффективно разделять ответственности между компонентами архитектуры MVC и улучшает сопровождаемость и расширяемость приложения.

Роли компонентов MVC в Laravel

В архитектуре MVC в Laravel компоненты играют особую роль и выполняют специфические функции, чтобы обеспечить эффективную разработку и управление проектом. Вот основные роли, которые играют компоненты MVC в Laravel:

Модель

Модель в Laravel отвечает за доступ и манипуляцию данными приложения. Она обеспечивает логику работы с базой данных, валидацию данных и другую бизнес-логику приложения. Модель является основой приложения и отвечает за хранение и получение данных.

Представление

Контроллер

Контроллер в Laravel отвечает за обработку запросов пользователя и взаимодействие с моделью и представлением. Он принимает запросы от пользователя и определяет, какая модель должна быть использована для получения данных и какое представление должно быть показано пользователю. Контроллер также выполняет валидацию данных и обрабатывает ошибки, если они возникают.

Таким образом, компоненты MVC в Laravel работают совместно, чтобы обеспечить разделение логики приложения, управление данными и представление результатов пользователю. Это позволяет разработчикам легко масштабировать и поддерживать проект, а также повышает читабельность и ясность кода.

Преимущества и недостатки

Преимущества архитектуры MVC в Laravel:

1. Разделение логики: Архитектура MVC позволяет разделить логику приложения на три основных компонента — модели, представления и контроллеры. Это позволяет разработчикам легко управлять кодом и делает приложение более организованным и поддерживаемым.

2. Повторное использование кода: MVC позволяет повторно использовать код, так как модели и контроллеры могут быть использованы в разных представлениях и на разных страницах приложения.

3. Улучшение безопасности: Архитектура MVC помогает обеспечить лучшую безопасность приложения, так как логика и данные разделены на разные компоненты. Это позволяет проводить проверку данных перед сохранением и предотвращает множество уязвимостей.

Недостатки архитектуры MVC в Laravel:

1. Сложность для начинающих: Использование MVC может быть сложным для начинающих разработчиков, так как требуется обучение и понимание концепций моделей, представлений и контроллеров. Это может затруднить начало работы с Laravel.

2. Дополнительный объем кода: Из-за разделения логики на три компонента, требуется написание дополнительного кода для обработки запросов, взаимодействия с базой данных и создания представлений. Это может увеличить объем кода в приложении.

3. Возможность перегружения приложения: Неправильное использование архитектуры MVC может привести к созданию сложных и запутанных моделей, представлений и контроллеров, что может замедлить работу приложения и привести к перегрузке.

Преимущества использования архитектуры MVC в Laravel

Разделение ответственности: MVC разделяет приложение на три основных компонента — модели, представления и контроллеры, что позволяет разработчикам лучше организовать и структурировать код и обеспечивает легкость поддержки и расширения проекта.

Улучшенная отказоустойчивость: MVC позволяет создавать независимые компоненты и их поочередное тестирование, что упрощает обнаружение и исправление ошибок в программном обеспечении.

Улучшенная скорость разработки: С использованием MVC в Laravel разработчику необходимо заботиться только о конкретной области — модели, представлении или контроллере, что сокращает время разработки и позволяет быстро прототипировать приложение.

Легкая поддержка: Разделение проекта на компоненты упрощает понимание и поддержку кода. Каждый компонент можно изменять или оптимизировать отдельно от остальных.

Масштабируемость: Использование архитектуры MVC делает проект более масштабируемым. Можно легко добавить новые функции и изменять существующие без влияния на другие компоненты.

В целом, использование архитектуры MVC в Laravel позволяет разрабатывать качественное, гибкое и легко поддерживаемое ПО.

Недостатки использования архитектуры MVC в Laravel

Несмотря на все преимущества, архитектура MVC в Laravel имеет свои недостатки, которые важно учитывать при разработке:

Сложность: MVC является довольно сложной архитектурой, особенно для новичков в Laravel. Вам потребуется время и усилия, чтобы полностью понять все компоненты и взаимодействие между ними.

Выбор ошибочной роли: Неправильный выбор роли для определенного компонента (модели, представления или контроллера) может привести к трудностям в разработке и поддержке кода.

Избыточность: Некоторые проекты могут быть слишком простыми для использования полной архитектуры MVC в Laravel. В этом случае, использование MVC может быть избыточным и увеличить сложность проекта.

Производительность: Использование MVC в Laravel может негативно сказаться на производительности приложения, особенно если проект большой и имеет множество запросов к базе данных.

Зависимость от фреймворка: Использование архитектуры MVC привязывает ваш проект к Laravel. Если в будущем вы решите переключиться на другой фреймворк, вам придется переписывать значительную часть кода.

* + 1. Встроенные инструменты (Eloquent ORM, Blade, Artisan, миграции)

Eloquent ORM

Система объектно-реляционного отображения (ORM) Eloquent - это реализация шаблона ActiveRecord в Laravel для работы с базами данных. Для каждой таблице в базе есть соответствующий класс-модель, который используется для работы с этой таблицей. Модель позволяет запрашивать данные из таблиц, вставлять новые записи и совершать другие действия.

Модель Eloquent предполагает, что каждая таблица имеет первичный ключ с именем id. Если название этого столбца другое, то необходимо передать свойство "primaryKey" (по аналогии со свойством "table), в котором указывается другое название столбца.

Модель Eloquent также предполагает, что первичный ключ является инкрементным числов и автоматически приводит его к типу int. Если нужно использовать другие типы ключей, то задайте свойству "incrementing" значение "false"

Blade

Blade — это мощный системный шаблонизатор, который входит в стандартный комплект фреймворка Laravel. Он предоставляет возможность использовать общепринятый синтаксис PHP в представлениях и добавляет удобные инструменты для работы с шаблонами, такие как наследование и секции. Blade специально разработан для упрощения и оптимизации процесса создания веб-страниц.

Blade отличается удобным и интуитивно понятным синтаксисом, который позволяет разработчикам быстро создавать шаблоны. К основным возможностям Blade можно отнести:

Использование директив: Blade расширяет возможности PHP с помощью директив, что позволяет выполнять условия, циклы и другие структуры управления прямо в шаблоне.

Наследование шаблонов: Поддерживает принцип «Don't Repeat Yourself» (DRY), предоставляя инструменты для наследования шаблонов и определения разделов.

Включение подшаблонов: Возможность вставить один шаблон в другой, что упрощает управление повторно используемыми элементами.

Передача данных в шаблон: Blade дает возможность легко передавать данные из контроллера в шаблон, что ускоряет процесс разработки.

Использование Blade в Laravel имеет множество преимуществ:

Чистый и понятный синтаксис: это делает код легким для понимания и поддержки.

Интеграция с PHP: шаблонизатор не отделяет PHP от своего синтаксиса, благодаря чему разработчики могут использовать всю мощь нативного PHP.

Удобство при работе с макетами: Blade упрощает процесс работы с макетами страниц, позволяя разработчикам сосредоточиться на важных аспектах приложения.

Адаптивность: Blade легко расширяется и адаптируется к нуждам проекта.

Artisan

В Laravel есть собственный интерфейс командной строки под названием Artisan. Он похож на командную строку Linux, но команды полезны для создания приложений Laravel. С помощью этого инструмента командной строки мы можем создавать модели, контроллеры, выполнять миграцию данных и многое другое. Сначала нам нужно изменить каталог в вашей командной строке (например, cmd в Windows или терминале в Linux/Mac) или в любом другом программном обеспечении командной строки на каталог вашего приложения Laravel.

Все команды artisan: затем, введя в консоли «php artisan list», мы сможем увидеть все команды artisan, которые можно использовать. Мы также можем ввести «php artisan» и нажать Enter, чтобы отобразить все доступные команды в консоли artisan. С помощью этой команды мы сможем увидеть длинный список команд, которые поддерживает php artisan.

Миграции.

Миграции в Laravel – способ управления изменениями в структуре базы данных вашего проекта. Они позволяют создавать и изменять таблицы, индексы, ограничения и другие элементы базы данных с помощью кода, а не с использованием SQL-запросов.

Миграции позволяют легко развертывать и поддерживать структуру базы данных, а также включать изменения в распределение вашего проекта с помощью системы контроля версий.

Они позволяют вам работать в команде, где каждый участник может иметь свою собственную копию базы данных, а миграции помогут вносить изменения и синхронизировать структуру базы данных между собой.

Создание миграций

Откройте командную строку и перейдите в корневую папку вашего проекта Laravel.

Выполните команду

php artisan make:migration create\_table\_name --create=table\_name

где

create\_table\_name – название миграции,

table\_name – название таблицы, которую вы хотите создать.

Laravel создаст новый файл миграции в папке database/migrations. Откройте его в текстовом редакторе.

В методе up() определите структуру таблицы, используя функции Laravel для создания столбцов, индексов, ограничений и других свойств.

Например, чтобы создать столбец name типа VARCHAR(255), вы можете использовать функцию $table->string (name, 255),

Также вы можете использовать другие функции для создания различных типов столбцов,для этого ознакомьтесь с документацией Laravel.

Для добавления индексов, внешних ключей или других ограничений, используйте соответствующие функции внутри метода Blueprint.

Например, чтобы добавить индекс на столбец email, вы можете использовать функцию $table->index (email).

Сохраните файл и закройте его.

Чтобы выполнить все непримененные миграции выполните команду:

php artisan migrate

Laravel выполнит миграцию и создаст указанную вами таблицу в базе данных.

Откат миграций

Откат позволяет отменить выполнение последних миграций и вернуть базу данных в предыдущее состояние.

Откройте командную строку и перейдите в корневую папку вашего проекта Laravel. Чтобы откатить последнюю миграцию выполните команду:

php artisan migrate:rollback

Laravel удалит из базы данных последнюю примененную миграцию и отменит изменения, сделанные на ней.

Вы также можете добавить флаг --step=N, где N – количество миграций для отката. Например, команда php artisan migrate:rollback --step=3 откатит три последние миграции.

Если вы хотите откатить все миграции, выполните команду:

php artisan migrate:reset

После применения команды Laravel вернет базу данных в начальное состояние.

Если вы хотите перекатить все миграции заново после отката, выполните команду:

php artisan migrate

Laravel повторно выполнит все миграции, создавая и обновляя таблицы базы данных в соответствии со структурой, определенной в миграциях.

Откат миграций в Laravel может быть полезен при разработке и тестировании проекта, когда требуется быстро вернуть базу данных в предыдущее состояние. Будьте осторожны при использовании этой команды на продакшене, чтобы избежать потери данных. Перед выполнением отката миграций рекомендуем создавать резервные копии базы данных.

**Глава 2. Практическая часть. Разработка интернет-магазина с использованием DDD.**

**2.1. Архитектура проекта**

#### ****2.2. Реализация доменной логики (на примере модуля Catalog)****

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### ****2.3. Работа с корзиной (модуль Cart)****

#### 

#### 

#### 

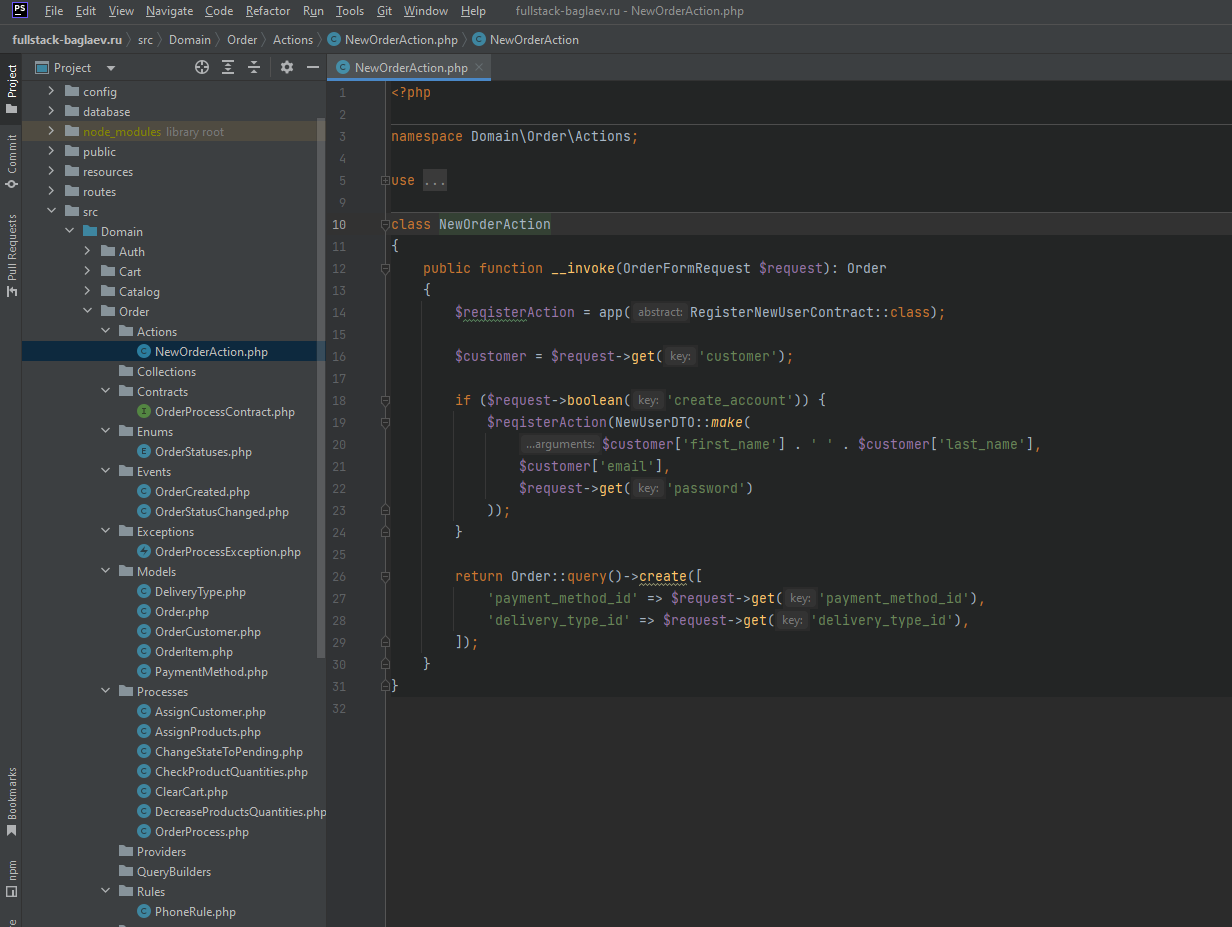
#### 

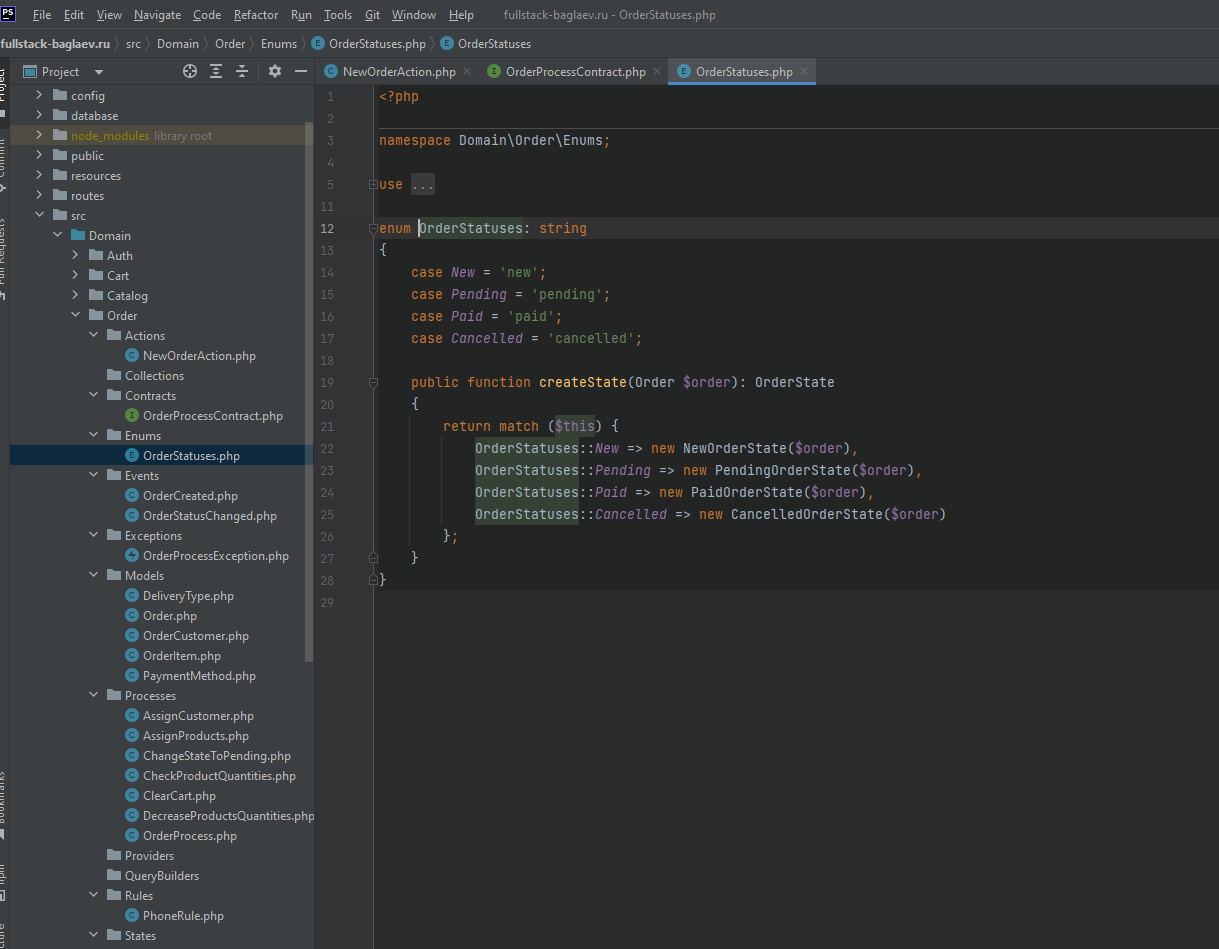
#### 

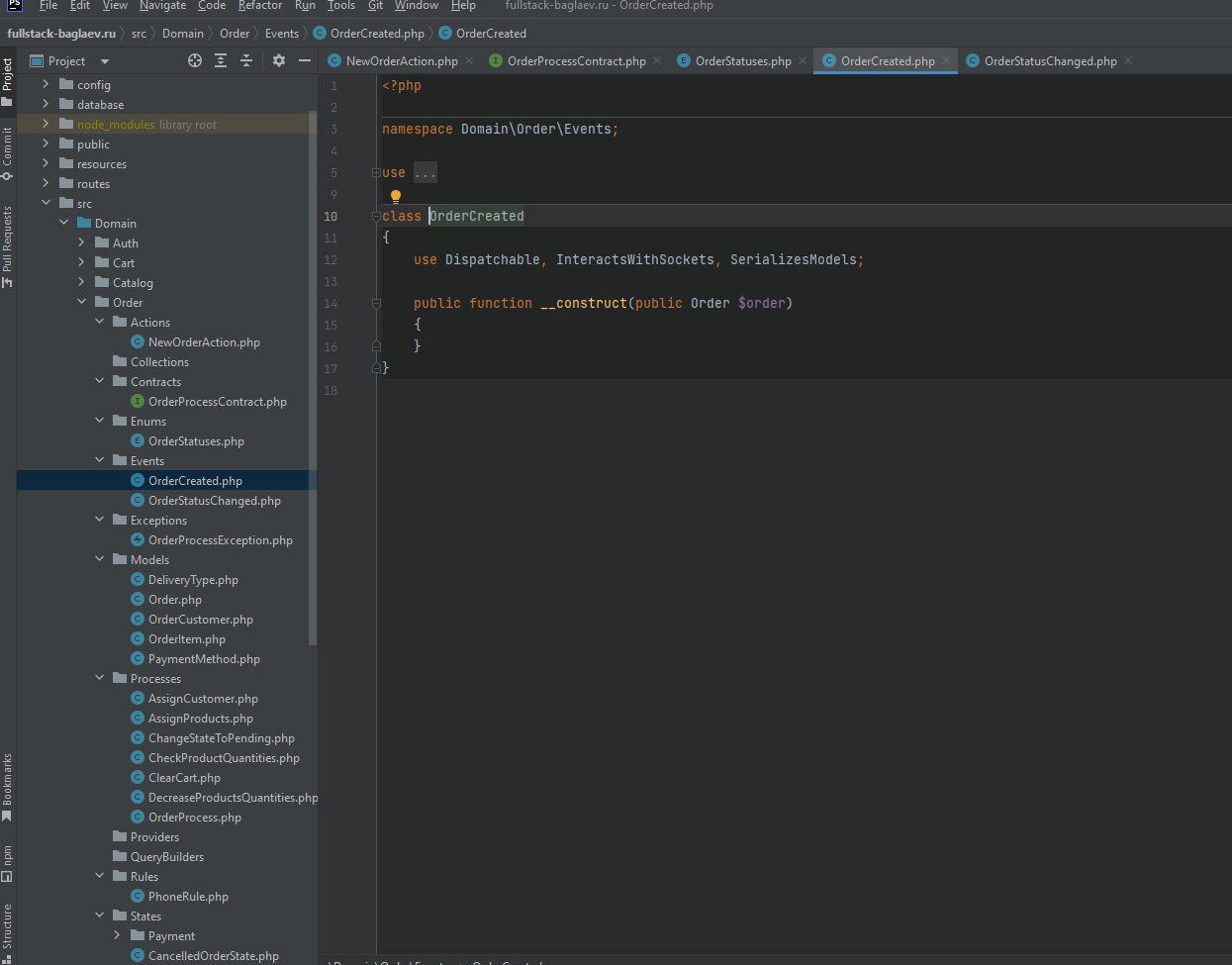
#### 

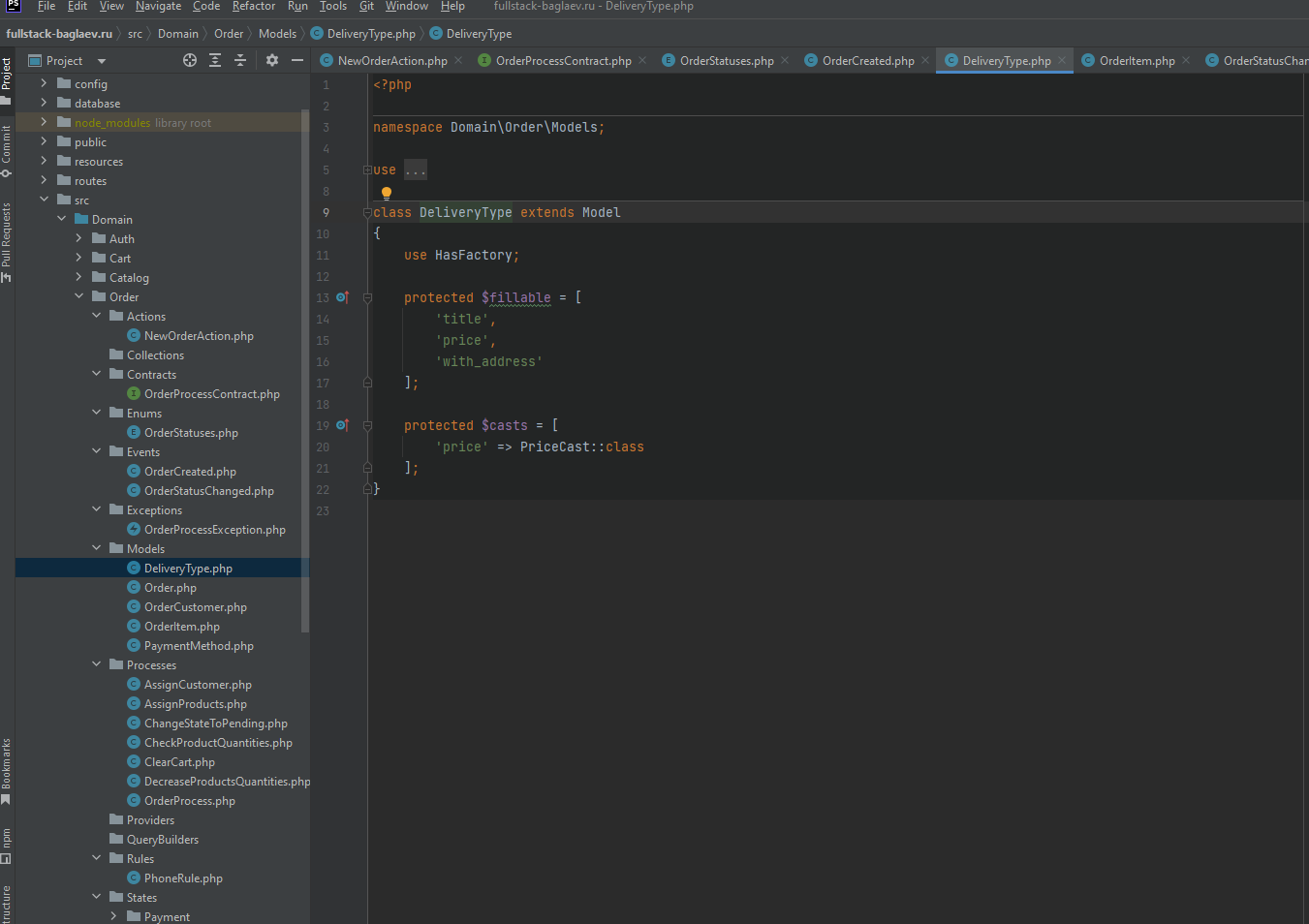
#### ****2.4. Оформление заказа (модуль Order)****

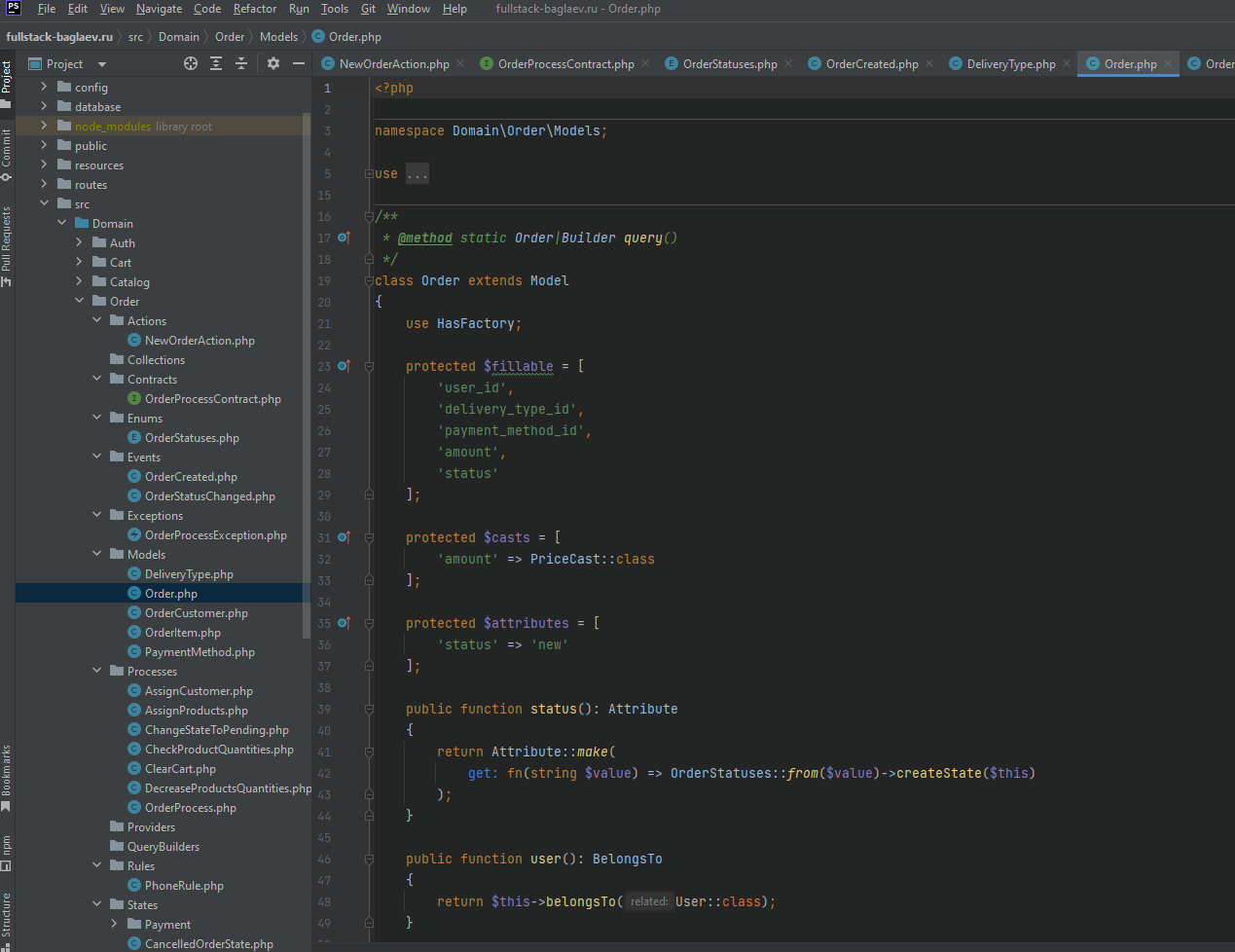
#### 

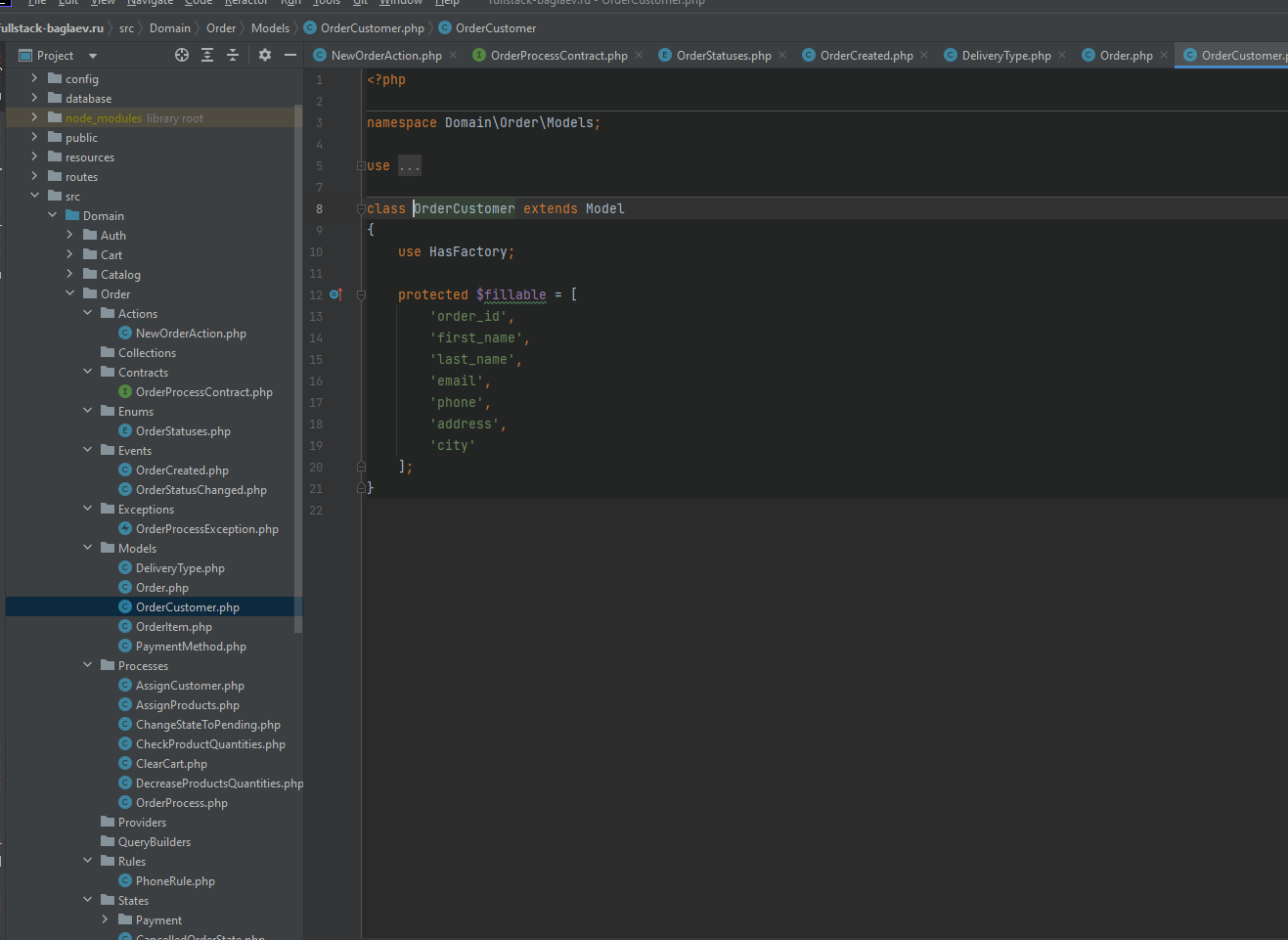


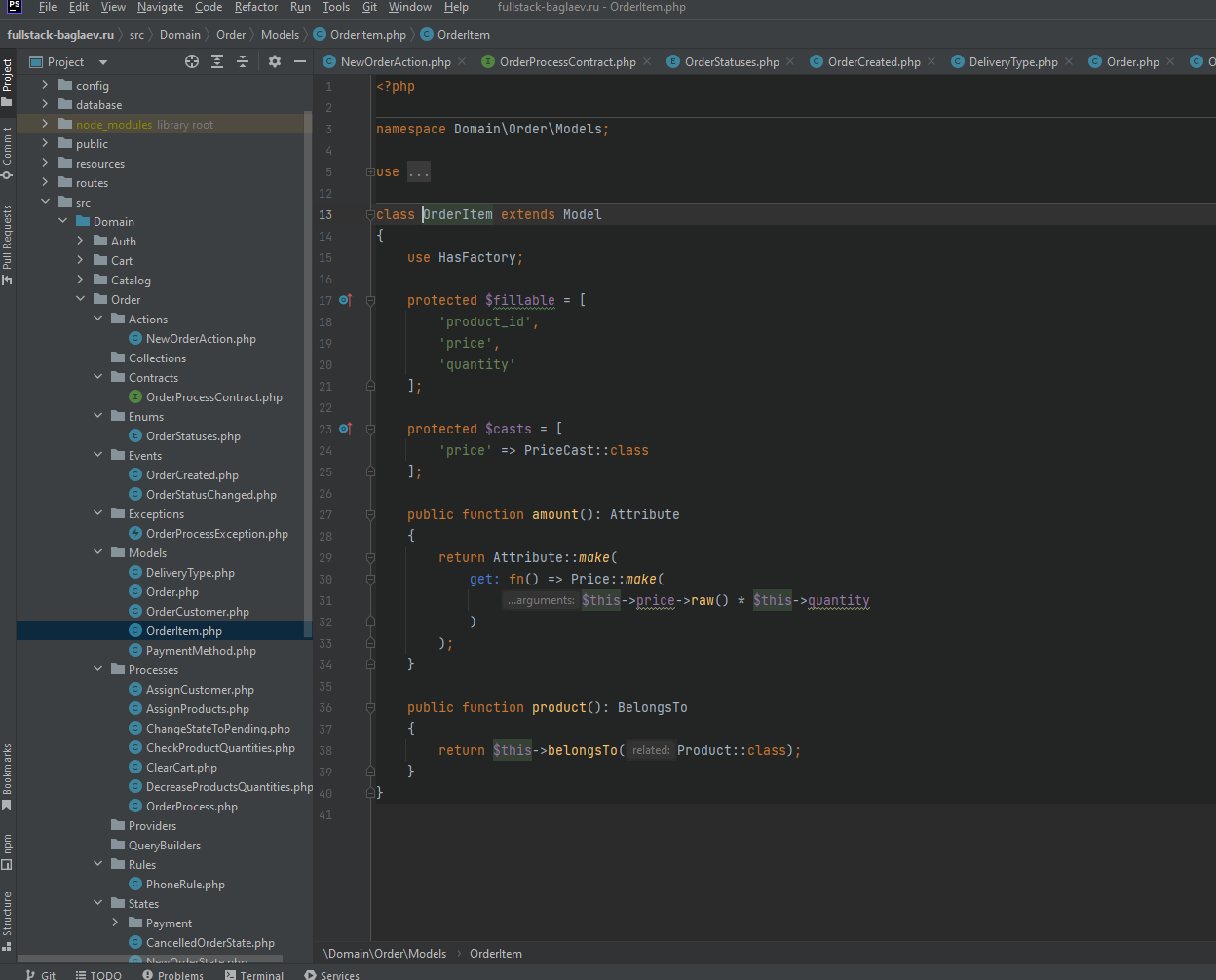












**Заключение**

В ходе выполнения дипломного проекта был разработан интернет-магазин на базе фреймворка Laravel с применением методологии Domain-Driven Design (DDD). Реализованы ключевые модули: каталог товаров, корзина, оформление заказа, система оплаты, управление пользователями и аналитика.

Использование DDD позволило четко разделить бизнес-логику на домены, что улучшило поддерживаемость и масштабируемость кода.

Проект протестирован на производительность, безопасность и отзывчивость интерфейса. В дальнейшем систему можно расширить за счет интеграции с CRM, добавления рекомендательных алгоритмов и мультиязычности.

Разработанный интернет-магазин соответствует современным требованиям к e-commerce и может быть успешно внедрен в коммерческую эксплуатацию.