

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АКАДЕМИЧЕСКИЙ КОЛЛЕДЖ

ОТЧЕТ
ПО УЧЕБНОЙ ПРАКТИКЕ
по профессиональному модулю
ПМ.03 Проектирование и разработка информационных
систем
ПМ.05 Сoadминистрирование и автоматизация баз данных и
серверов
программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование
период с «06» мая по «02» июня 2023 года

Наименование предприятия: ФГБОУ ВО «ВВГУ», Академический колледж,
г. Владивосток

Студент гр. С-ИП-21-1

Руководитель практики
от предприятия

Отчет защищен:
с оценкой

5

Руководитель
практики от ОО

Д.А. Атабаева

М.А. Федянин

Д.А. Атабаева

Д.А. Атабаева

Владивосток 2023

Содержание

Введение	3
1 Анализ деятельности ООО НПФ «ЭФФЕКТОВЕНТ»	4
1.1 Общие сведения	4
1.2 План мероприятий по проектированию и разработке интернет-магазина	5
2 Анализ и выбор инструментальных средств для проектирования и разработки интернет-магазина	7
3 Проектирование интернет-магазина	10
3.1 Проектирование интернет-магазина с помощью унифицированного языка моделирования	10
3.2 Проектирование интерфейса пользователя	13
4 Разработка интернет-магазина	19
4.1 Разработка основной логики программной и интерфейсной части интернет-магазина	19
4.2 Разработка личного кабинета пользователя	20
4.3 Выпуск и установка SSL-сертификата для интернет-магазина	21
5 Разработка базы данных интернет-магазина	23
5.1 Разработка и развертка базы данных на хостинге	23
5.2 Разработка генератора тестовых данных для тестирования работоспособности отображения информации из базы данных	26
5.3 Тестирование API интернет-магазина с помощью Postman	27
Заключение	31
Список используемых источников	32

Введение

В период с 06.05.2023 года по 02.06.2023 года, была пройдена учебная практика в организации ФГБОУ ВО «ВВГУ», Академический колледж, расположенном по адресу: Приморский край, г. Владивосток, ул. Гоголя, 41.

Основная цель учебной практики повышение компетенций по профессиональным модулям:

- 1) ПМ.03 «Проектирование и разработка информационных систем»;
- 2) ПМ.05 «Сoadминистрирование и автоматизация баз данных и серверов».

Для достижения целей необходимо решить следующие задачи:

- проанализировать деятельность предприятия ООО НПФ «ЭФФЕКТОВЕНТ»;
- проанализировать существующие интернет-магазины по выбранной тематике;
- провести анализ и выборку инструментальных средств для проектирования и разработки интернет-магазина;
- создать карту навигации;
- спроектировать интернет-магазин;
- разработать интернет-магазин;
- разработать проект базы данных интернет-магазина;
- разработать и развернуть базу данных на хостинге и интегрировать её в серверную часть интернет-магазина;
- обеспечить базовую безопасность базы данных и настроить резервное копирование данных для базы данных интернет-магазина;
- разработать генератор тестовых данных для тестирования работоспособности отображения информации из базы данных на страницах интернет-магазина;
- оформить тестовую документацию для API интернет-магазина и провести тестирование API с помощью Postman;
- выпустить и установить SSL-сертификат для интернет-магазина;
- сбор и обработка материалов, необходимых для составления отчета по практике.

Учебная практика необходимо для получения первичных профессиональных умений и навыков. Заключается в практической подготовке будущего специалиста и закрепляет полученные теоретические знания.

1 Анализ деятельности ООО НПФ «ЭФФЕКТОВЕНТ»

1.1 Общие сведения

Фирма «ЭФФЕКТОВЕНТ» создана в 1992 году группой инженеров-специалистов в области строительства и эксплуатации климатических систем зданий. За время своего существования компания приобрела значительный опыт создания и управления микроклиматом, как отдельных групп помещений, так и целых зданий.

Основные направления деятельности предприятия: разработка, проектирование, поставка оборудования, монтаж, наладка, техническое обслуживание следующих систем: отопление, вентиляция и кондиционирование воздуха; водоснабжение и канализация; тепло- и электроснабжение (в том числе автономное); электрооборудование и электроосвещение; автоматизация и диспетчеризация управления инженерными системами; учет тепловой энергии, теплоносителей, расхода воды.

Особенностью предлагаемых ООО НПФ «ЭФФЕКТОВЕНТ» климатических систем является относительно низкие затраты на их реализацию и на последующую эксплуатацию.

Являясь одним из лидеров на Дальневосточном рынке по поставкам профессионального климатического оборудования, фирма «ЭФФЕКТОВЕНТ» предлагает заказчикам лучшее европейское и американское оборудование, прошедшее сертификацию ISO 9001 [1].

Организационная структура предприятия (рисунок 1.1)

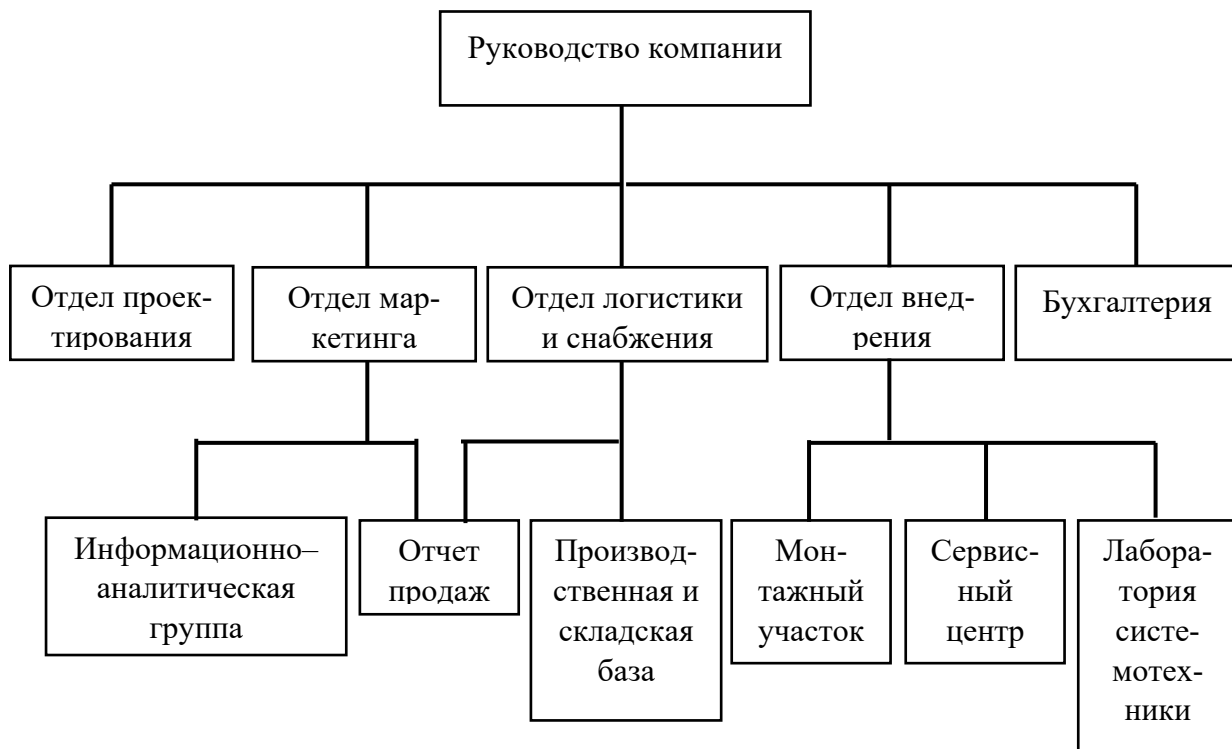


Рисунок 1.1 – Организационная структура предприятия

Перечислим несколько причин, почему интернет-магазин необходим для данного предприятия:

1. Расширение рынка.

Интернет-магазин предоставляет возможность предприятию достичь новых клиентов и рынков. Он позволит привлечь потенциальных заказчиков из разных регионов, а также за пределами Дальнего Востока. Это даст предприятию возможность увеличить свою клиентскую базу и повысить объем продаж.

2. Удобство и доступность.

Интернет-магазин предоставляет удобную и доступную платформу для клиентов «ЭФФЕКТОВЕНТ». Заказчики смогут легко найти и приобрести необходимое климатическое оборудование прямо через веб-сайт. Они смогут ознакомиться с широким ассортиментом продукции, описаниями и характеристиками, а также сделать заказ в любое удобное для них время.

3. Повышение осведомленности о продукции.

Интернет-магазин может служить важным инструментом для информирования потенциальных клиентов о предлагаемом «ЭФФЕКТОВЕНТ» оборудовании и услугах. Интернет-магазин может предоставить детальные описания продукции, техническую информацию, руководства по установке и использованию. Это поможет потенциальным клиентам принять информированное решение о покупке и установке климатических систем.

4. Повышение конкурентоспособности.

Интернет-магазин позволит предприятию быть на шаг впереди конкурентов. В условиях современного рынка все больше людей предпочитают совершать покупки онлайн. Интернет-магазин даст предприятию преимущество перед конкурентами и привлечет новых клиентов.

1.2 План мероприятий по проектированию интернет-магазина

Проектирование и разработка интернет-магазина – это сложный процесс, который включает в себя множество этапов. План мероприятий по проектированию и разработки интернет-магазина включает в себя следующее [2]:

1. Анализ рынка и конкурентов.

Необходимо проанализировать существующие интернет-магазины в данной нише, изучить их особенности, преимущества и недостатки. Это поможет выявить конкурентное преимущество и определить основные требования к будущему интернет-магазину.

2. Разработка концепции интернет-магазина.

Необходимо определить основные параметры будущего интернет-магазина, такие как ассортимент, цены, дизайн, удобство использования, методы доставки и оплаты, система управления заказами и т.д.

3. Проектирование и создание дизайна.

Необходимо создать дизайн интернет-магазина, который будет соответствовать заданным параметрам и удобству использования для пользователей.

4. Разработка и интеграция функционала.

Необходимо разработать необходимый функционал интернет-магазина, такой как корзина, оформление заказа и т.д. При необходимости можно также интегрировать сторонние решения.

5. Тестирование и отладка.

Необходимо протестировать интернет-магазин на различных устройствах и браузерах, а также отладить все функции и процессы для обеспечения бесперебойной работы интернет-магазина.

6. Релиз и продвижение.

После завершения разработки и тестирования, необходимо выпустить интернет-магазин в релиз и провести продвижение для привлечения целевой аудитории.

Важно отметить, что каждый из этих этапов требует тщательного планирования и тщательной работы. Кроме того, весь процесс разработки интернет-магазина должен быть выполнен с учетом сроков и бюджета проекта. Чем более тщательно подойти к планированию и организации работ на каждом этапе, тем больше шансов на успех проекта.

2 Анализ инструментальных средств для проектирования и разработки интернет-магазина

Для проектирования информационной системы стоит выбрать средство проектирования. Одним из таких выступают сайты для построения UML-диаграмм.

UML – унифицированный язык моделирования (Unified Modeling Language) – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем [3].

Проанализируем четыре известных сайта для построения UML-диаграмм с их преимуществами и недостатками:

1) draw.io – бесплатный и open-source инструмент, поддерживающий множество типов диаграмм, включая UML. Имеет облачную версию, а также можно скачать приложение для использования офлайн. Один из главных недостатков – не всегда удобный интерфейс;

2) Lucidchart – инструмент для создания различных типов диаграмм, включая UML. Имеет бесплатную версию, а также расширенные версии с дополнительными функциями за определенную плату. Недостаток – не все функции доступны в бесплатной версии;

3) Creately – инструмент, который позволяет создавать диаграммы и схемы, включая UML. Имеет бесплатную версию, а также платные версии с расширенными функциями. Недостаток – некоторые функции доступны только в платной версии;

4) PlantUML – инструмент для создания диаграмм на основе текстовых описаний, включая UML. Бесплатен и open-source, не требует установки. Недостаток – не всегда удобный для работы с графическими элементами.

Каждый из этих сайтов имеет свои преимущества и недостатки. Однако, на мой взгляд, draw.io представляется самым универсальным и удобным из перечисленных, так как он бесплатный, open-source и поддерживает широкий спектр типов диаграмм.

Для разработки информационной системы стоит выбрать средство разработки.

Одним из таких выступают, так называемые, фреймворки, то есть наборы предопределенных инструментов, библиотек и структур, которые облегчают разработку приложений.

Проанализируем четыре известных фреймворков для разработки информационной системы интернет-магазина на языке программирования Python с их преимуществами и недостатками:

1. Django – это высокоуровневый фреймворк, который содержит в себе множество готовых инструментов и библиотек для создания веб-приложений, в том числе и интернет-магазинов. Преимуществами Django включают: удобство и скорость разработки благодаря готовому функционалу и широкой документации; большое количество плагинов и библиотек,

которые могут быть использованы для расширения функционала; встроенная система администрирования, которая позволяет управлять данными и пользователями приложения. Недостатками Django включают: из-за множества готовых инструментов, приложение может быть слишком тяжелым и медленным; Django требует достаточно большого объема знаний и времени для его изучения и освоения.

2. Flask – это легковесный фреймворк для разработки веб-приложений на Python. Преимуществами Flask включают: простота и легкость в изучении и использовании; гибкость и возможность настройки для каждого проекта; отличная поддержка веб-сокетов. Недостатками Flask включают: Flask не имеет множества готовых инструментов и библиотек, как Django; Flask может быть слишком легковесным для сложных проектов.

3. Pyramid – это фреймворк, который позволяет разработчикам создавать веб-приложения любой сложности. Преимуществами Pyramid включают: гибкость и возможность настройки под каждый проект; легкость в использовании и расширении функционала; отличная поддержка для RESTful API. Недостатками Pyramid включают: не имеет такого большого сообщества, как Django и Flask; не имеет готовых инструментов для администрирования приложений, как в Django.

4. Bottle – это очень легковесный фреймворк, который может использоваться для создания маленьких приложений и прототипов. Преимуществами Bottle включают: очень прост в использовании и имеет небольшой размер; обладает минимальными зависимостями и хорошей документацией. Недостатком Bottle заключается в том, что он может не подойти для более сложных проектов из-за его ограниченной функциональности.

Какой фреймворк нужен для разработки информационной системы зависит от конкретных потребностей проекта. Для небольшого прототипа подойдет Bottle. Если нужно создать большое веб-приложение, то Django подходит для этого. Фреймворки Flask и Pyramid находятся между ними.

Рассмотрим технические средства для размещения информации в магазине, обеспечивающие администрирование и хранение файлов. Операционная система для рабочего места администратора интернет-магазина не имеет значения, потому что все управление будет реализовываться с помощью браузера.

Есть много подходов к размещению сайтов в сети. Размещение сайта на сервере в интернете напрямую зависит от возложенных на него задач. Есть вариант платного размещения и платы за сервер.

При размещении информационной системы, не требующей материальных затрат, владелец проекта очень ограничен в ресурсах и возможностях.

При плате за использование сервера возможности информационной системы будет зависеть от выбранного тарифного плана.

Выделяют следующие виды бесплатного размещения проекта на сервере:

- общий сервер;
- индивидуальное размещение;
- виртуальный выделенный сервер.

Общий сервер предполагает разделение имеющихся ресурсов между всеми пользователями в равной степени, без учета назначений сайта.

Индивидуальное размещение является возможностью одновременно обслуживать большое количество сайтов, что доступно многим пользователям и идеально подходит для небольших и средних сайтов, хотя и с наличием некоторых запретов на использование ресурсов сервера.

Выделенный сервер также называется физическим размещением, когда обслуживание сайта происходит на определенном сервере и его ресурсы доступны для использования только владельцу одного сайта, что позволяет обеспечить стабильную работу развернутых порталов, которые посещают сотни тысяч людей ежедневно.

Говоря о разрабатываемом проекте, стоит выбрать индивидуальное размещение, так как рассматриваемый интернет-магазин является средним по масштабам проектом, тем более в рамках учебной практики не планируется со временем привлекать большое количество покупателей к интернет-магазину, а значит нет необходимости в самых передовых функций.

По результатам анализа тремя несомненными лидерами рынка можно считать фреймворки Django, Flask и Pyramid. Тем не менее, принимая во внимание технические требования оптимальной реализации проекта интернет-магазина, стоит использовать фреймворк Django, так как интернет-магазин не будет являться маленьким веб-приложением, а соответственно для такой задачи подходит именно этот фреймворк.

3 Проектирование интернет-магазина

3.1 Проектирование интернет-магазина с помощью унифицированного языка моделирования

Перечислим основные UML-диаграммы, необходимые для проектирования интернет-магазина:

1. Диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей [3].

В данной информационной системе можно выделить следующие группы пользователей:

- 1) клиенты;
- 2) администраторы.

Каждая из групп пользователей может пользоваться нашей информационной системой по-своему. Клиенты могут:

- регистрироваться или авторизоваться под своей учетной записью;
- добавлять товар в корзину;
- оформлять заказ.

Администраторы могут:

- регистрироваться или авторизоваться под своей учетной записью;
- редактировать базу данных;
- добавлять данные о новом товаре в базу данных;
- удалять данные о товаре в базе данных.

В терминологии UML, человек в диаграмме называется актёром (англ. "actor"). В общем случае, актёр обозначает любые сущности, использующие систему. Этими сущностями могут быть люди, технические устройства или даже другие системы.

Изобразим актеры для групп пользователей, которые могут пользоваться нашей системой (рисунок 3.1)



Рисунок 3.1 – Группы пользователей данной информационной системы

Каждая группа пользователей использует определённые функции системы. На диаграмме вариантов использования функция системы изображается эллипсом, внутри которого записывается имя функции в форме глагола с пояснительными словами. Перечислим все типы функций системы:

- 1) отношение ассоциации – отображение на диаграмме информацию о том, какие варианты использования могут быть использованы каждым актёром;
- 2) отношение обобщения – означает, что некоторый актёр (вариант использования) может быть обобщён до другого актёра (варианта использования). Стрелка направлена от частного случая (специализации) к общему случаю;
- 3) отношение включения – показывает, что некоторый вариант использования включает в себя другой вариант использования в качестве составной части;
- 4) отношение расширения – выборочное отношение включения. Если отношение включения обозначает, что элемент обязательно включается в состав другого элемента, то в случае отношения расширения это включение необязательно.

Как итог, наша диаграмма принимает следующий вид (рисунок 4.2)

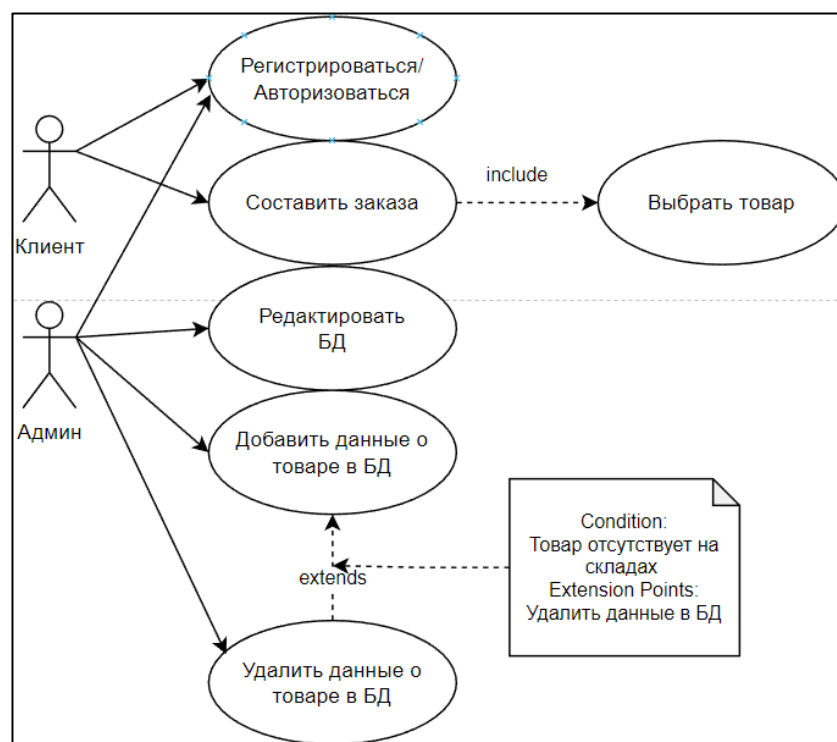


Рисунок 3.2 – Итоговый вид диаграммы вариантов использования

В данной диаграмме показаны группы пользователей и функции, которые каждая группа может использовать в данной информационной системе.

2. Диаграмма последовательности (англ. sequence diagram) – UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл

объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента.

Основными элементами диаграммы последовательности являются взаимодействующие объекты с различными стереотипами и связи между ними – взаимодействующие объекты обмениваются между собой некоторой информацией.

Элемент Actor может использоваться для представления пользователя, инициирующего поток событий.

Каждый объект имеет пунктирную линию, называемую "линией жизни", где этот элемент существует и потенциально принимает участие во взаимодействиях. Фокус управления обозначается прямоугольником на линии жизни объекта [3].

Как итог, наша диаграмма принимает следующий вид (рисунок 3.3)

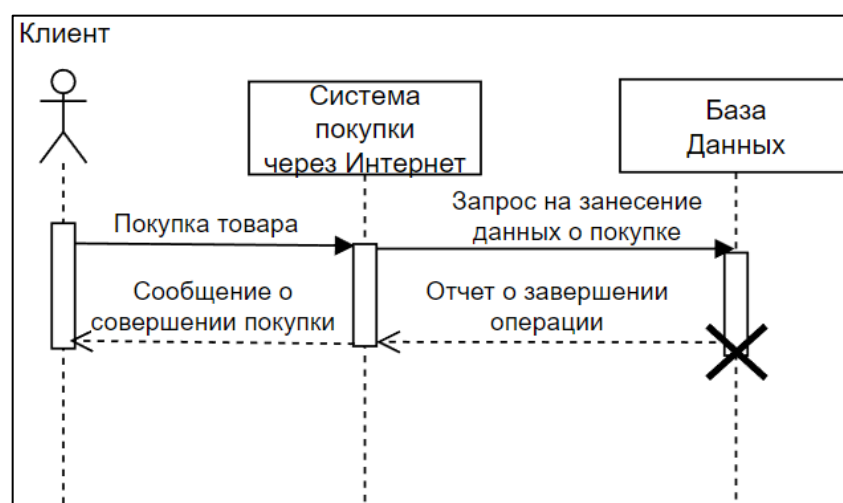


Рисунок 3.3 – Итоговый вид диаграммы последовательности

Рассмотрим итоговую диаграмму. Клиент здесь является актером, который инициирует поток событий. Клиент может взаимодействовать с системой покупки через Интернет, то есть совершать покупку через сайт, а та, в свою очередь, может взаимодействовать с базой данных. В данной диаграмме показан процесс покупки товара пользователем и вывод сообщения пользователю о совершении покупки.

3. Диаграмма классов (англ. class diagram) – UML-диаграмма, где наборы объектов с одинаковыми атрибутами, операциями, связями и семантикой описываются с помощью классов. Графически класс изображается в виде прямоугольника, разделенного на три блока горизонтальными линиями [3]:

- 1) имя класса – это текстовая строка;
- 2) атрибут (свойство) класса – это именованное свойство класса, описывающее диапазон значений, которые может принимать экземпляр атрибута;
- 3) операция (метод) класса – это реализация метода класса.

Ко всему прочему, в диаграмме классов есть связи, которые представляют собой базовые строительные блоки для описания отношений в UML, используемые для разработки хорошо согласованных моделей. Существуют четыре типа связей в диаграмме классов:

- 1) зависимость – представляет собой связь между двумя элементами модели, в которой изменение одного элемента может привести к изменению семантики другого элемента;
- 2) ассоциация – показывает, что объекты одного класса связаны с объектами другого класса таким образом, что можно перемещаться от объектов одного класса к другому;
- 3) обобщение – выражает специализацию или наследование, в котором потомок строится по спецификациям родителя;
- 4) реализация – это семантическая связь между классами, когда один из них (поставщик) определяет соглашение, которого второй (клиент) обязан придерживаться.

Как итог, наша диаграмма принимает следующий вид (рисунок 3.4)

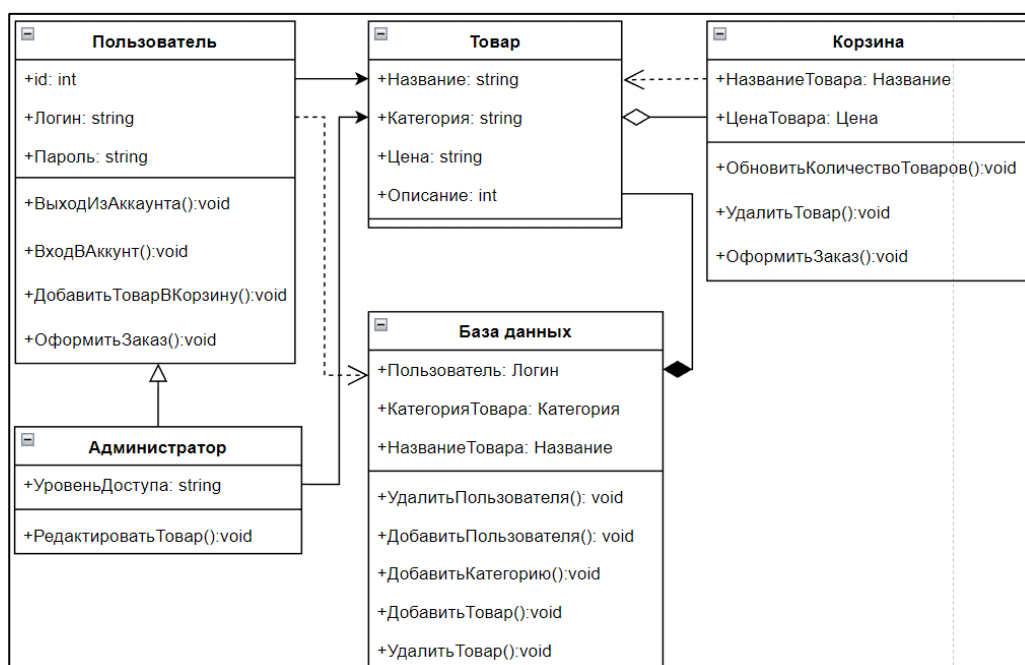


Рисунок 3.4 – Итоговый вид диаграммы классов

В данной диаграмме показаны классы и связи, с помощью которых одни классы могут взаимодействовать с другими классами и обмениваться между ними информацией.

3.2 Проектирование интерфейса пользователя

Рассмотрим проектирование интерфейса пользователя с помощью двух компонентов, а именно:

1. Карта навигации – это диаграмма, которая показывает структуру сайта и помогает пользователям легко ориентироваться в нем. Она может включать в себя основные разделы, подразделы и страницы сайта, а также связи между ними.

Рассмотрим карту навигации рассматриваемого интернет-магазина (рисунок 3.5)

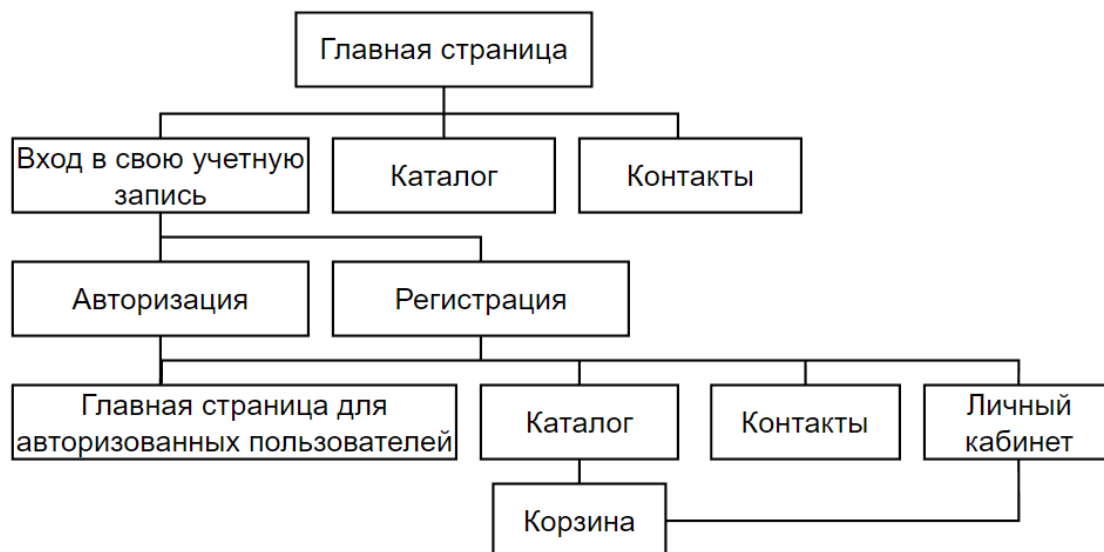


Рисунок 3.5 – Карта навигации интернет-магазина

В данной карте навигации показаны разделы, доступные различным пользователям в зависимости от роли.

Неавторизованным пользователям доступны следующие разделы: главная страница, кнопки авторизации и регистрации, контакты и каталог без возможности добавления товара в корзину (кнопка корзины тоже недоступна).

Авторизованным пользователям после авторизации под свою учетную запись будут доступны следующие разделы: главная страница, контакты, кнопка для открытия личного кабинета, кнопка выхода из своей учетной записи, каталог с возможностью добавления товара в корзину, кнопка корзины.

2. Макеты страниц для интернет-магазина – это графические модели, которые демонстрируют расположение элементов на страницах сайта. Макеты страниц помогают определить, как будут располагаться основные элементы сайта, такие как логотип, меню, поисковая строка, изображения, текст и кнопки [4].

Рассмотрим макеты страниц и расположение их элементов:

1. Главная страница. Макет главной страницы интернет-магазина представлен на рисунке 3.6.

Логотип интернет-магазина расположен в левом верхнем углу страницы, меню навигации расположена в верхней части страницы, кнопка авторизации расположена в правом верхнем углу страницы, кнопка регистрации расположена в правом верхнем углу страницы, название предприятия и краткое описание его деятельности расположены на главной странице, общие сведения о предприятии расположены на главной странице, контактные данные предприятия расположены в нижней части страницы.



Рисунок 3.6 – Макет главной страницы интернет-магазина

2. Контакты. Макет блока «Контакты» интернет-магазина представлен на рисунке 3.7.

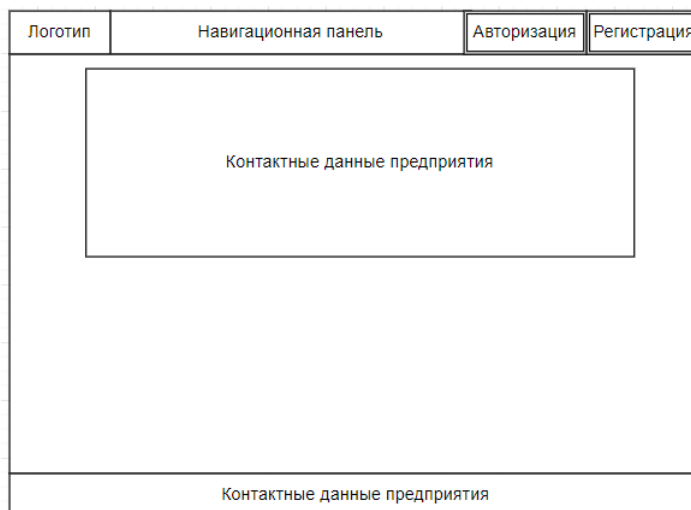


Рисунок 3.7 – Макет блока «Контакты» интернет-магазина

Логотип расположен в левом верхнем углу страницы, меню навигации расположена в верхней части страницы, кнопка авторизации расположена в правом верхнем углу страницы, кнопка регистрации расположена в правом верхнем углу страницы, контактные данные предприятия расположены на главной странице и в нижней части страницы.

3. Каталог. Макет каталога интернет-магазина представлен на рисунке 3.8.

Логотип расположен в левом верхнем углу страницы, меню навигации расположена в верхней части страницы, кнопка авторизации расположена в правом верхнем углу страницы, кнопка регистрации расположена в правом верхнем углу страницы, поле поиска товара по названию расположено на главной странице, список категорий расположен в средней части страницы, страница товаров расположена в средней части страницы, кнопка пагинации и контактные данные предприятия расположены в нижней части страницы.

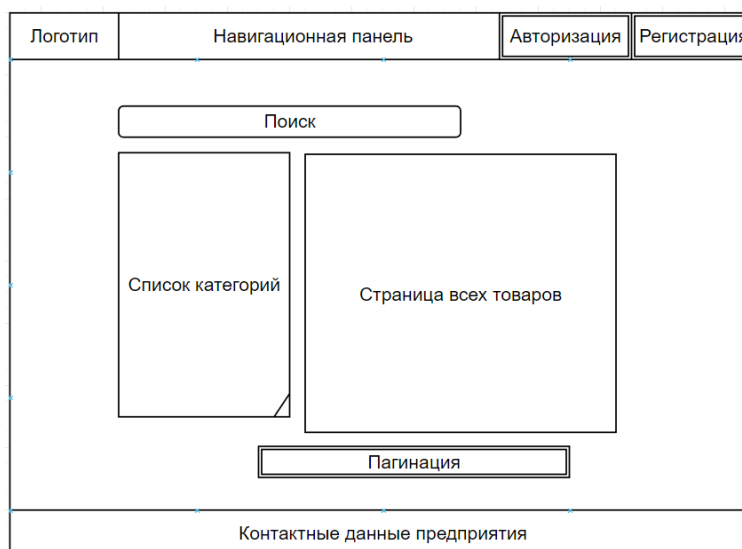


Рисунок 3.8 – Макет каталога интернет-магазина

4. Корзина. Макет корзины интернет-магазина представлен на рисунке 3.9.

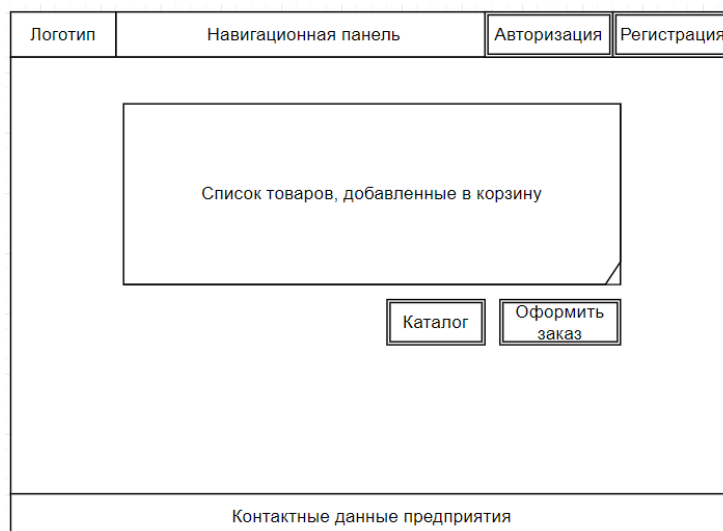


Рисунок 3.9 – Макет корзины интернет-магазина

Логотип расположен в левом верхнем углу страницы, меню навигации расположена в верхней части страницы, кнопка авторизации расположена в правом верхнем углу страницы, кнопка регистрации расположена в правом верхнем углу страницы, список добавленных в корзину товаров расположен на главной странице, кнопка каталога расположена в средней части страницы, кнопка оформления заказа расположена в средней части страницы, контактные данные предприятия расположены в нижней части страницы.

5. Личный кабинет. Макет личного кабинета интернет-магазина представлен на рисунке 3.10.

Логотип расположен в левом верхнем углу страницы, меню навигации расположена в верхней части страницы, кнопка авторизации расположена в правом верхнем углу страницы,

кнопка регистрации расположена в правом верхнем углу страницы, личная информация пользователя расположена на главной странице, кнопка корзины расположена на главной странице, контактные данные предприятия расположены в нижней части страницы.

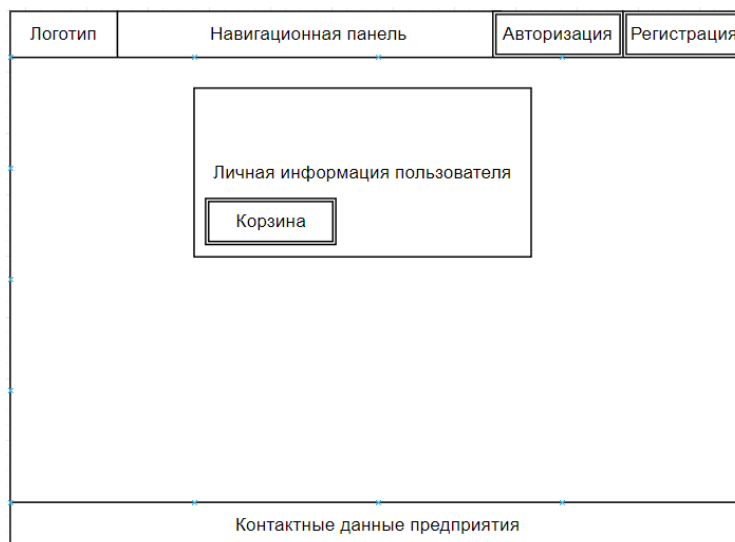


Рисунок 3.10 – Макет личного кабинета интернет-магазина

6. Форма регистрации. Макет регистрации интернет-магазина представлен на рисунке 3.11.

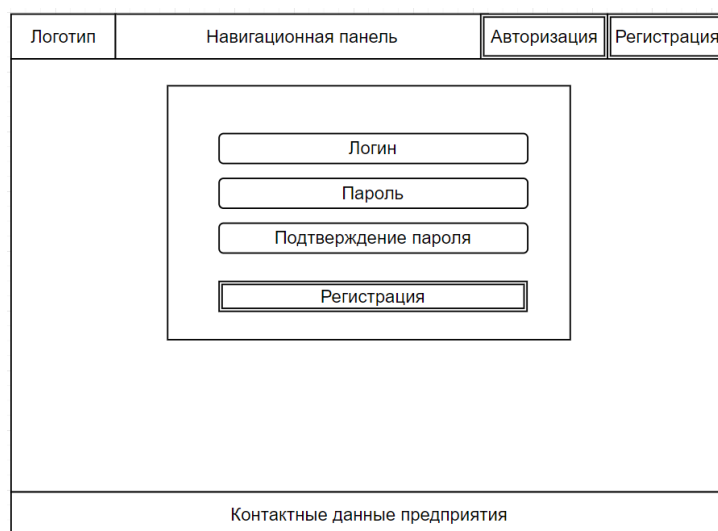


Рисунок 3.11 – Макет регистрации интернет-магазина

Логотип расположен в левом верхнем углу страницы, меню навигации расположена в верхней части страницы, кнопка авторизации расположена в правом верхнем углу страницы, кнопка регистрации расположена в правом верхнем углу страницы, поле для ввода логина расположено на главной странице, поле для ввода пароля расположено на главной странице, поле для ввода подтвержденного пароля расположено на главной странице, кнопка регистрации

расположена на главной странице, контактные данные предприятия расположены в нижней части страницы.

7. Форма авторизации. Макет авторизации интернет-магазина представлен на рисунке 3.12.

Логотип	Навигационная панель	Авторизация	Регистрация
Логин Пароль Авторизация Авторизация через ВКонтакте			
Контактные данные предприятия			

Рисунок 3.12 – Макет авторизации интернет-магазина

Логотип расположен в левом верхнем углу страницы, меню навигации расположена в верхней части страницы, кнопка авторизации расположена в правом верхнем углу страницы, кнопка регистрации расположена в правом верхнем углу страницы, поле для ввода логина расположено на главной странице, поле для ввода пароля расположено на главной странице, кнопки авторизации и авторизации через ВКонтакте расположены на главной странице, контактные данные предприятия расположены в нижней части страницы.

4 Разработка интернет-магазина

4.1 Разработка основной логики программной и интерфейсной части интернет-магазина

Основную логику с применением фреймворка Django интернет-магазина можно разделить на несколько компонентов, включая аутентификацию пользователей, управление товарами, функциональность корзины и управление заказами. Django предоставляет набор инструментов, которые облегчают создание этих компонентов [5].

Одним из первых шагов в разработке интернет-магазина является создание функциональности аутентификации пользователей. Django предоставляет встроенную систему аутентификации пользователей. Эта система позволяет пользователям создавать учетную запись, входить и выходить с сайта. Django также обеспечивает поддержку управления паролями.

После создания системы аутентификации пользователей, следующим шагом будет создание функциональности управления продуктом. Это включает создание схемы базы данных для хранения информации о продукте, такой как название, описание, цена и изображение. Django предоставляет объектно-реляционный маппер (ORM), который позволяет разработчикам определять модели баз данных с помощью классов Python. Эти модели можно использовать для создания, чтения, обновления и удаления (CRUD) информации о продукте.

После создания функциональности управления продуктом следующим шагом будет создание функциональности корзины. Это включает в себя предоставление пользователям возможности добавлять товары в корзину, просматривать содержимое корзины и оформлять заказ. Django предоставляет фреймворк сессий, который позволяет разработчикам хранить данные о сессии пользователя, включая его корзину.

Помимо основной логики интернет-магазина, Django можно использовать для создания интерфейса интернет-магазина. Интерфейс включает в себя дизайн и макет сайта, а также навигацию и взаимодействие с пользователем. Django предоставляет несколько инструментов для создания интерфейса интернет-магазина.

Одним из наиболее важных инструментов для создания интерфейса интернет-магазина является движок шаблонов Django. Этот механизм позволяет разработчикам создавать HTML-шаблоны, которые могут быть использованы для отображения содержимого сайта. Шаблоны могут включать динамическое содержимое.

Django предоставляет поддержку для создания форм, которые можно использовать для сбора данных от пользователей. Эти формы можно использовать для создания формы регистрации, формы входа в систему или формы оформления заказа. Django предоставляет несколько встроенных полей формы, таких как текстовые поля, поля выбора и поля даты.

Наконец, Django предоставляет поддержку для создания представлений (views), которые представляют собой функции Python, обрабатывающие входящие запросы и генерирующие ответы. Эти представления можно использовать для отображения шаблонов, обработки форм и взаимодействия с базой данных.

4.2 Разработка личного кабинета пользователя

С помощью фреймворка Django рассмотрим разработку личного кабинета пользователя. Личный кабинет позволяет пользователю просмотреть имя своей учетной записи, дата создания учетной записи, а также кнопка корзины, которая позволяет открыть страницу корзины. Но, чтобы пользователь смог открыть свой личный кабинет, необходимо авторизоваться под своей учетной записью.

Рассмотрим следующие шаги для разработки личного кабинета пользователя:

1. Определяем модель User. Django предоставляет встроенную модель User, которая может быть использована для хранения информации о пользователе. Модель User поставляется с полями для хранения имени пользователя, пароля, адреса электронной почты, имени и фамилии. Если необходимо хранить дополнительную информацию, например, адрес доставки или номер телефона пользователя, можно создать пользовательскую модель User, расширив встроенную модель User.

2. Создаем форму регистрации.

Форма регистрации необходима для того, чтобы пользователи могли создать новый аккаунт. Django предоставляет встроенную UserCreationForm, которую можно использовать для создания простой регистрационной формы.

3. Создаем форму авторизации.

Форма авторизации необходима для того, чтобы пользователи могли войти в свой аккаунт. Django предоставляет встроенную AuthenticationForm, которую можно использовать для создания простой формы входа.

4. Создаем представление личного кабинета пользователя.

Представление личного кабинета пользователя необходимо для того, чтобы пользователи могли просматривать личную информацию. Это представление можно создать, добавив новую функцию в python-файле views.py, которая будет отображать шаблон личного кабинета.

Рассмотрим python-файлы, которые используются для создания личного кабинета пользователя:

- 1) models.py – этот файл определяет модель User, которая используется для хранения информации о пользователе;

- 2) forms.py – этот файл определяет формы регистрации и авторизации;

3) `views.py` – этот файл определяет представления, которые используются для отображения страниц регистрации, авторизации, личного кабинета пользователя;

4) `urls.py` – этот файл определяет шаблоны URL для представлений регистрации, авторизации, личного кабинета пользователя.

Итоговый вид личного кабинета пользователя представлен на рисунке 4.1.

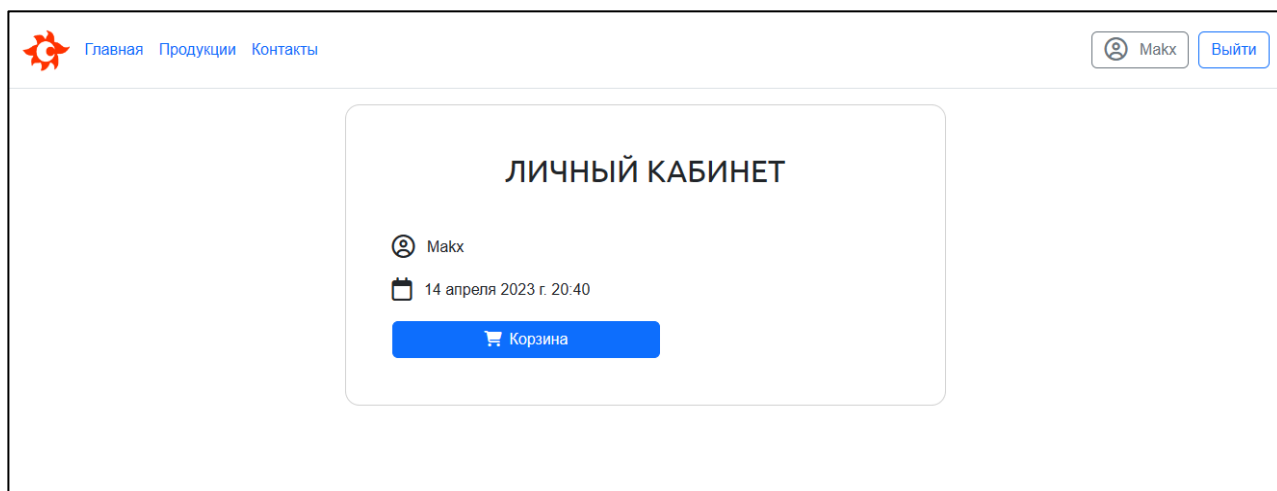


Рисунок 4.1 – Личный кабинет пользователя

Таким образом, фреймворк Django предоставляет простой способ создания личного кабинета пользователя для интернет-магазина. Определив модель `User`, создав формы регистрации и авторизации, определив представления и шаблоны URL, можно создать функциональную систему пользовательского аккаунта.

4.3 Выпуск и установка SSL-сертификата для интернет-магазина

SSL-сертификат – это криптографический протокол, который обеспечивает безопасность данных пользователей, подтверждение прав собственности на сайт, предотвращение создания поддельной версии сайта злоумышленниками и обеспечивает доверие со стороны пользователей [6].

Рассмотрим, как выпустить самоподписанный SSL-сертификат для интернет-магазина, расположенного на локальном компьютере. Самоподписанный SSL-сертификат отличается от других тем, что выпускается он на локальном компьютере и, обычно, браузеры не доверяют таким сертификатам.

Для выпуска самоподписанного SSL-сертификата воспользуемся простой утилитой `mkcert`. Перед этим необходимо установить данную утилиту на компьютере с помощью командной оболочки Windows PowerShell с правами администратора. Дальнейшие действия будут проводиться в том же Windows PowerShell. После установки утилита `mkcert`

автоматически создает и устанавливает локальный центр сертификации в корневом хранилище системы, а также генерирует сертификаты, которым доверяют локально.

Далее необходимо в директории проекта Django в терминале ввести следующую команду “python manage.py runserver_plus 0.0.0.0:8000 --cert-file .\localhost.crt”, установив перед этим пакет Python django_extensions и указав в python-файле settings.py в разделе INSTALLED_APPS. Эта команда запустит интернет-магазин на локальном компьютере с использованием SSL-сертификата.

Результат запуска интернет-магазина с использованием SSL-сертификата представлен на рисунке 4.2.

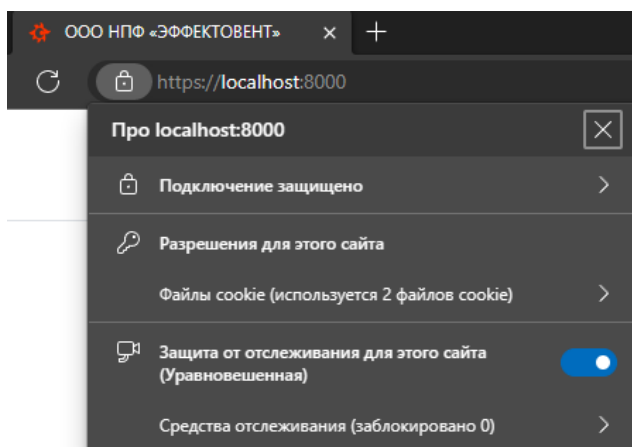


Рисунок 4.2 – Результат запуска интернет-магазина с использованием SSL-сертификата

Подробные сведения самоподписанного SSL-сертификата представлены на рисунке 4.3.



Рисунок 4.3 – Подробные сведения самоподписанного SSL-сертификата

Был выпущен и установлен самоподписанный SSL-сертификат, который обеспечивает безопасное подключение для интернет-магазина, расположенного на локальном компьютере.

5 Разработка базы данных интернет-магазина

5.1 Разработка и развертка базы данных на хостинге Timeweb

База данных для интернет-магазина была разработана с помощью SQLite3.

SQLite3 – это популярная встроенная система управления реляционными базами данных, которая обеспечивает легковесное, быстрое и автономное хранение данных. Это программная библиотека, которая реализует транзакционный SQL-движок и позволяет пользователям взаимодействовать с базами данных с помощью SQL-запросов [7].

Помимо SQLite3, Django поддерживает множество баз данных, включая такие популярные базы данных, как PostgreSQL, MySQL и Oracle.

В Django можно определять схему базы данных с помощью моделей. Модель – это класс Python, который представляет собой таблицу базы данных. Каждый атрибут класса модели соответствует полю базы данных.

В рамках рассматриваемого интернет-магазина было создано две модели, то есть две таблицы базы данных: Категории и Товары. Также была использована встроенная модель User, которая хранит информацию о пользователях и необходима для управления пользователями, аутентификации, авторизации и других связанных задач.

Схема базы данных интернет-магазина была автоматически спроектирована на основе файла db.sqlite3 с помощью бесплатного программного обеспечения DbSchema (рисунок 5.1)

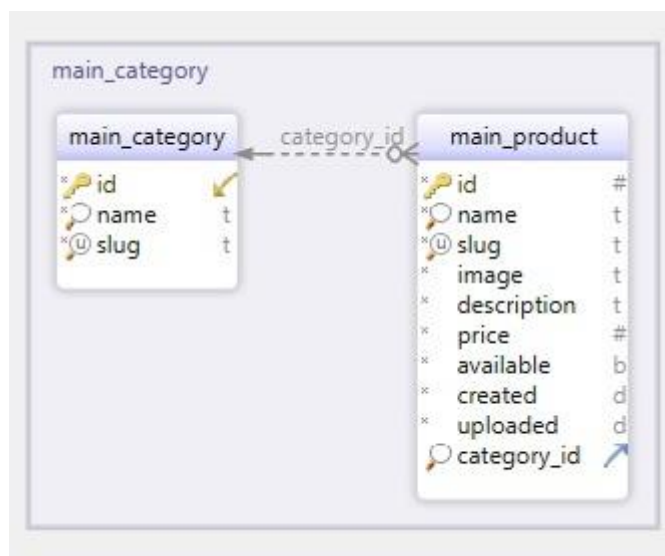


Рисунок 5.1 – Схема базы данных интернет-магазина

Рассмотрим вышеописанную схему базы данных. Она состоит из двух таблиц:

1. main_category – это таблица, которая хранит информацию о категориях товаров. Рассмотрим поля данной таблицы:

1) id – уникальный идентификатор категории;

2) name – название категории;

3) slug – ссылка категории.

2. main_product – это таблица, которая хранит информацию о товарах. Рассмотрим поля данной таблицы:

1) id – уникальный идентификатор товара;

2) name – название товара;

3) slug – ссылка товара;

4) image – изображение товара;

5) description – описание товара;

6) price – цена товара;

7) available – наличие товара, с помощью этого поля можно отображать товары на страницах интернет-магазина, или нет;

8) created – дата внесения товара в базу данных;

9) uploaded – дата изменения какой-либо информации о товаре в базе данных;

10) category_id – категория товара.

В таблице main_category и main_product поля id являются первичными ключами. В таблице main_product поле category_id является внешним ключом.

Между полем id из таблицы main_category и полем category_id из таблицы main_product существует связь «один-ко-многим».

Рассмотрим развёртку базы данных на хостинге. Но, перед этим необходимо развернуть сам интернет-магазин на хостинге. Для развёртки интернет-магазина на хостинге был использован российский хостинг Timeweb.

Timeweb – это российский хостинг, отличающийся стабильным функционированием, безопасностью, быстродействием и множеством предложений для размещений сайтов. Он имеет идеально структурированную панель администрирования и обеспечивает моментальный доступ к интернет-площадке. Timeweb предоставляет следующие услуги:

1) виртуальный хостинг;

2) регистрация доменов;

3) SSL-сертификаты;

4) выделенные сервера;

5) конструктор сайтов.

Для развёртки рассматриваемого проекта достаточно воспользоваться виртуальным хостингом. Timeweb предлагает различные тарифы и разница между тарифами в том, что чем дороже тариф, тем больше бонусов и возможностей открывается для пользователей. Выбираем любой тариф, так как после выбора Timeweb предоставляет пользователям тестовый

период, в течение которого пользователи могут попробовать выбранный им тариф и, если их устраивает выбранный тариф, приобрести его.

После выбора тарифа пользователя встречает панель управления хостингом.

Панель управления хостингом представлена на рисунке 5.2.

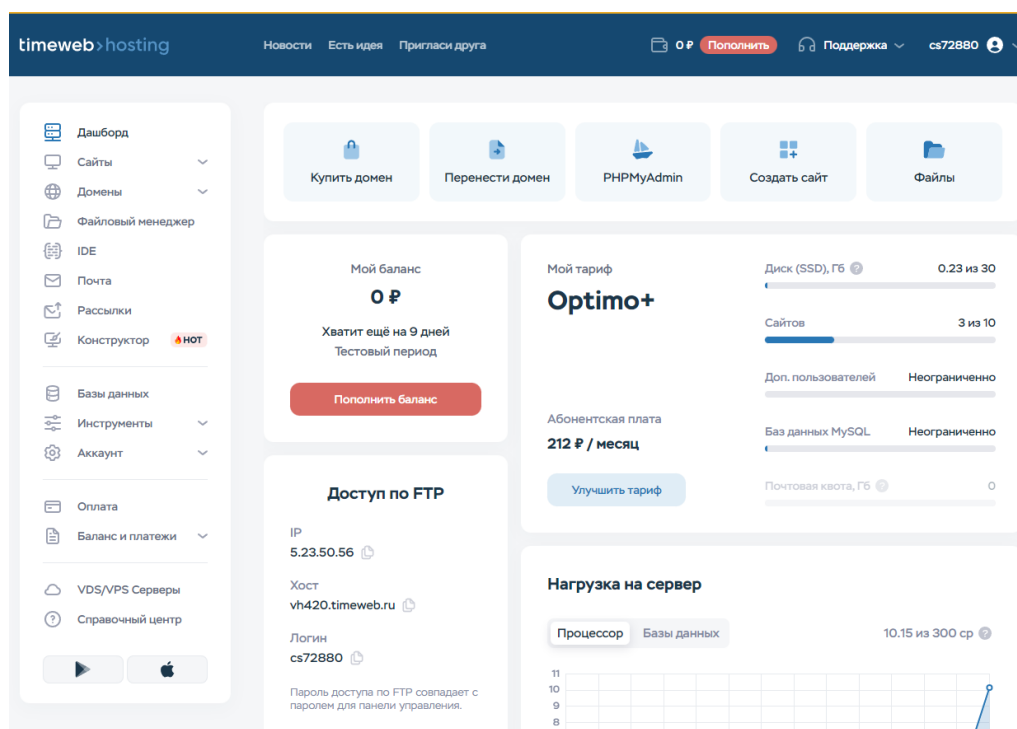


Рисунок 5.2 – Панель управления хостингом в Timeweb

В разделе «Дашборд» необходимо активировать SSH и с помощью программного обеспечения PuTTY подключиться к серверу. Теперь можно установить виртуальное окружение на сервере. Инструкцию по установке, созданию и активации виртуального окружения можно найти в справочном центре Timeweb.

После активации виртуального окружения необходимо установить Django на сервере и создать проект Django. После создаем файл .htaccess в директории сайта (public_html/) и редактируем файл wsgi.py. Коды для этих файлов можно найти в справочном центре Timeweb.

После вышеперечисленных действий теперь можно перенести все необходимые папки и файлы django-проекта, расположенном на локальном компьютере, в django-проект на хостинге. Далее запускаем сайт и видим, что интернет-магазин отображается, но не отображаются стили и картинки. Какие команды следует использовать в терминале для отображения статического контента на сайте описаны в справочном центре Timeweb.

Сайт интернет-магазина почти готов. Он корректно отображается, также корректно отображаются стили и картинки, далее необходимо развернуть базу данных на хостинге. Для этого необходимо перенести файл базы данных db.sqlite3 в django-проект на хостинге. После,

в терминале, создаем таблицы в базе данных на хостинге. Команда для создания таблиц в базе данных на хостинге можно узнать в справочном центре Timeweb.

Чтобы интернет-магазин стал общедоступным, то есть сайт могли открыть другие пользователи, необходимо зарегистрировать доменное имя в разделе «Мои домены» и привязать его к сайту интернет-магазина. Timeweb предлагает данную услугу, однако услуга не бесплатная, но без этой услуги сайт интернет-магазина будет доступен и отображаться только администратору.

5.2 Разработка генератора тестовых данных для тестирования работоспособности отображения информации из базы данных

Рассмотрим разработку генератора тестовых данных, который необходим для тестирования работоспособности отображения информации из базы данных интернет-магазина.

Для разработки генератора тестовых данных создадим функцию в python-файле test.py, в качестве аргумента функции будет являться число товаров, которое нужно сгенерировать и сохранить в базу данных интернет-магазина. Таким образом, созданные генератором товары будут отображаться в интернет-магазине.

Код генератора тестовых данных для тестирования работоспособности отображения информации из базы данных представлен на листинге 5.1.

```
from .models import Product
import random
import string
def generate_products(num_products):
    for i in range(num_products):
        name = ".join(random.choices(string.ascii_uppercase + string.digits, k=10))
        slug = name.lower()
        description = ".join(random.choices(string.ascii_uppercase + string.digits, k=1000))
        price = round(random.uniform(10.0, 1000.0), 2)
        Product.objects.create(name=name, slug=slug, description=description, price=price)
```

Листинг 5.1 – Код генератора тестовых данных

Для запуска кода необходимо ввести в терминале следующую команду “python manage.py shell”, которая запустит интерактивную командную оболочку Python, позволяющая выполнить Python-код в контексте Django-приложения. Далее импортируем функцию командой “from main.tests import generate_products” и, наконец, вызываем функцию generate_products() для генерации тестовых данных.

5.3 Тестирование API интернет-магазина с помощью Postman

API – это набор определенных правил и протоколов, которые определяют, как различные компоненты программного обеспечения могут взаимодействовать друг с другом. API предоставляет интерфейс для взаимодействия с функциональностью или данными программы, позволяя разработчикам использовать и интегрировать эти компоненты в свои собственные приложения [8].

Разработаем тестовую документацию, предназначенная для проверки функциональности и корректности работы API интернет-магазина. Рассмотрим все запросы, описания тестовых сценариев, входных данных и ожидаемых выходных данных с помощью программного обеспечения Postman:

1. Получение списка пользователей.

Данный запрос возвращает список всех пользователей, в том числе отображает уникальный идентификатор пользователя, имя пользователя, наличие прав суперпользователя, а также список групп, в которых пользователь входит.

REST – `https://localhost:8000/api/users/` (GET-запрос)

Тестовые сценарии:

- запрос списка пользователей;
- проверка соответствия полученных данных ожидаемым значениям.

Ожидаемый ответ в формате JSON представлен на листинге 5.2.

```
{
  "id": 1,
  "username": "Makx",
  "is_superuser": true,
  "groups": []
}
```

Листинг 5.2 – Ожидаемый ответ в формате JSON

2. Получение списка товаров.

Данный запрос возвращает список всех товаров, в том числе отображает категорию, название, описание и цену каждого товара.

REST – `https://localhost:8000/api/products/` (GET-запрос)

Тестовые сценарии:

- запрос списка товаров;
- проверка соответствия полученных данных ожидаемым значениям.

Ожидаемый ответ в формате JSON представлен на листинге 5.3.

```
[{
  "category": 4,
  "name": "VCA-18HRN (C)*",
  "slug": "vca-18hrn-s",
  "description": "Мощность, кВт: холод - 5.3  тепло - 6  потр. - 1,83
Расход воздуха, м3/ч: 860",
  "price": "1631.00"
}, {
  "category": 4,
  "name": "Product 2",
  "slug": "product-2",
  "description": "Lorem ipsum dolor sit amet, consectetur adipiscing",
  "price": "876.00"
}]
```

Листинг 5.3 – Ожидаемый ответ в формате JSON

3. Получение списка категорий.

Данный запрос возвращает список всех категорий.

REST – <https://localhost:8000/api/categories/> (GET-запрос)

Ожидаемый ответ в формате JSON представлен на листинге 5.4.

```
[{
  "name": "Кондиционеры Venterra",
  "slug": "kondicionery-venterra"
}, {
  "name": "Крышные вентиляторы",
  "slug": "kryshnye-ventilyatory"
}]
```

Листинг 5.4 – Ожидаемый ответ в формате JSON

Тестовые сценарии:

- запрос списка категорий;
- проверка соответствия полученных данных ожидаемым значениям.

4. Регистрация пользователя.

Данный запрос регистрирует нового пользователя в системе.

REST – <https://localhost:8000/api/users/> (POST-запрос)

Входные данные в формате JSON представлен на листинге 5.5.

```
{ "username": "Macco",  
  "is_superuser": false,  
  "groups": [] }
```

Листинг 5.5 – Входные данные в формате JSON

Ожидаемый ответ в формате JSON представлен на листинге 5.6.

```
{ "id": 40,  
  "username": "Macco",  
  "is_superuser": false,  
  "groups": [] }
```

Листинг 5.6 – Ожидаемый ответ в формате JSON

Тестовые сценарии:

- попытка зарегистрировать нового пользователя с уникальными данными;
- попытка зарегистрировать пользователя с уже существующим именем.

5. Создание категории.

Данный запрос создает новую категорию товаров.

REST – <https://localhost:8000/api/categories/> (POST-запрос)

Входные данные в формате JSON представлен на листинге 5.7.

```
{  
  "name": "New Category",  
  "slug": "new-category"  
}
```

Листинг 5.7 – Входные данные в формате JSON

Ожидаемый ответ в формате JSON представлен на листинге 5.8.

```
{  
  "name": "New Category",  
  "slug": "new-category"  
}
```

Листинг 5.8 – Ожидаемый ответ в формате JSON

Тестовые сценарии:

- попытка создания категории с уникальным названием и slug;

– попытка создания категории с уже существующим названием или slug.

6. Создание товара.

Данный запрос создает новый товар.

REST – <https://localhost:8000/api/products/> (POST-запрос)

Входные данные в формате JSON представлен на листинге 5.9.

```
{
  "category": 1,
  "name": "New Product",
  "slug": "new-product",
  "description": "Lorem ipsum dolor sit amet",
  "price": "59.00"
}
```

Листинг 5.9 – Входные данные в формате JSON

Ожидаемый ответ в формате JSON представлен на листинге 5.10.

```
{
  "category": 1,
  "name": "New Product",
  "slug": "new-product",
  "description": "Lorem ipsum dolor sit amet",
  "price": "59.00"
}
```

Листинг 5.10 – Ожидаемый ответ в формате JSON

Тестовые сценарии:

- попытка создания товара с уникальным названием и slug;
- попытка создания товара с уже существующим slug.

При тестировании API интернет-магазина на основе вышеперечисленной тестовой документации была выполнена проверка функциональности и корректности работы API интернет-магазина, были рассмотрены все запросы, описания тестовых сценариев, входных данных и ожидаемых выходных данных с помощью программного обеспечения Postman.

Заключение

В результате прохождения учебной практики все поставленные задачи были выполнены, закреплены знания и навыки работы с разработкой и проектированием информационной системы для интернет-магазина на основе деятельности предприятия ООО НПФ «ЭФФЕКТОВЕНТ» соответственно индивидуальному заданию.

Во время прохождения учебной практики были решены следующие задачи:

- проанализировать деятельность предприятия ООО НПФ «ЭФФЕКТОВЕНТ»;
- проанализировать существующие интернет-магазины по выбранной тематике;
- провести анализ и выборку инструментальных средств для проектирования и разработки интернет-магазина;
- создать карту навигации;
- спроектировать интернет-магазин;
- разработать интернет-магазин;
- разработать проект базы данных интернет-магазина;
- разработать и развернуть базу данных на хостинге и интегрировать её в серверную часть интернет-магазина;
- обеспечить базовую безопасность базы данных и настроить резервное копирование данных для базы данных интернет-магазина;
- разработать генератор тестовых данных для тестирования работоспособности отображения информации из базы данных на страницах интернет-магазина;
- оформить тестовую документацию для API интернет-магазина и провести тестирование API с помощью Postman;
- выпустить и установить SSL-сертификат для интернет-магазина;
- сбор и обработка материалов, необходимых для составления отчета по практике.

Была разработана интернет-магазина на основе деятельности предприятия ООО НПФ «ЭФФЕКТОВЕНТ» с использованием языков программирования Python, HTML, CSS и JavaScript с применением фреймворка Django.

Список используемых источников

- 1 О компании [Электронный ресурс] // effectovent.ru. – URL: <http://effectovent.ru/companу/> (дата обращения: 09.05.2023)
- 2 Азарин, А. Создание интернет-магазина своими руками. Без воды и лишних слов. – Москва: Издательство Питер, 2018. – 288 с.
- 3 Винарский, Я.С. Web-аппликации в интернет-маркетинге: проектирование, создание и применение: практическое пособие / Я.С. Винарский, Р.Д. Гутгарц. – Москва: ИНФРА-М, 2023. – 269 с. + Доп. Материалы [Электронный ресурс]. – (Среднее профессиональное образование). – Текст: электронный. – URL: <https://znanium.com/catalog/product/1891781> (дата обращения: 18.05.2023).
- 4 Роббинс Дж. Н., Саттон Л. Веб-дизайн. Основы проектирования и разработки / Перевод с англ. – Москва: ДМК Пресс, 2019. – 416 с.
- 5 Архангельский А. Django. Подробное руководство / А. Архангельский. – Москва: Издательство "Питер", 2019. – 672 с.
- 6 Зубишин С.А Роль SSL сертификатов в безопасной передаче данных / С.А. Зубишин // Молодой исследователь Дона: ДГТУ. – 2021. – №3. – С. 32-36.
- 7 Бахтияров И.Р. Технологический стек Django + Bootstrap 5 + Sqlite как инструмент для ускоренной разработки веб-приложения. И.Р. Бахтияров – Самара: СГУПС, 2022. – 50 с.
- 8 Гончаров А. API. Основы разработки приложений. Архитектура, протоколы, безопасность / А. Гончаров. – Санкт-Петербург: Питер, 2018. – 352 с.

Индивидуальное задание по учебной практике

Студент _____ Федянин Максим Андреевич _____,
ФИО

обучающийся на 2 курсе по специальности 09.02.07 «Информационные системы и программирование» прошел учебную практику в объеме 147 часов с «6» мая 2023 г. по «2» июня 2023 г.

в организации ФГБОУ ВО "ВВГУ", Академический колледж, г. Владивосток
наименование организации, юридический адрес

Виды и объем работ в период учебной практики

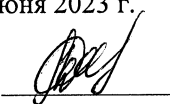
№ п/п	Вид работ	Кол-во часов
1	Установочная лекция, инструктаж по охране труда и технике безопасности, распределение по рабочим местам	4
2	Разработать план мероприятий для этапов проектирования и разработки интернет-магазина.	2
3	Проанализировать существующие интернет-магазины по выбранной тематике.	6
4	Провести анализ и выборку инструментальных средств для проектирования и разработки.	6
5	Создать карту навигации для интернет-магазина. На карте в зависимости от специфики системы выделить разделы, доступные различным пользователям в зависимости от роли, описать условия перехода из различных разделов (при необходимости).	6
6	Спроектировать интернет-магазин. В отчете привести: - диаграмму вариантов использования; - диаграмму последовательности; - диаграмму классов; - макеты страниц.	12
7	Разработать интернет-магазин, который должен соответствовать следующим требованиям: - наличие навигационной панели; - наличие удобного каталога и сегментация по параметрам; - возможность осуществления заказа с сайта; - наличие личного кабинета пользователя и корзины, которая запоминается даже при выходе со страницы; - статистика просмотра каждого товара, а также количества заказов; - наличие блока «Контакты»; - приятный UI и удобный UX.	36
8	Разработать проект базы данных интернет-магазина.	6
9	Разработать и развернуть базу данных на хостинге и интегрировать её в серверную часть интернет-магазина.	18
10	Обеспечить базовую безопасность базы данных.	6
11	Настроить резервное копирование данных для базы данных интернет-магазина.	9
12	Разработать генератор тестовых данных для тестирования работоспособности отображения информации из базы данных на страницах интернет-магазина.	9
13	Оформить тестовую документацию для API интернет-магазина с подробным описанием тестовых сценариев, входных и выходных данных.	6

<i>№ n/n</i>	<i>Вид работ</i>	<i>Кол-во часов</i>
14	Провести тестирование API интернет-магазина с помощью Postman.	6
15	Выпустить и установить SSL-сертификат для интернет-магазина.	3
16	Обобщить материалы практики, оформить отчет и необходимые документы по практике.	12

Дата выдачи задания «3» мая 2023 г.

Срок сдачи отчета по практике «2» июня 2023 г.

Подпись руководителя практики



/Атабаева Д.А., преподаватель АК ВВГУ

АТТЕСТАЦИОННЫЙ ЛИСТ

Студент Федянин Максим Андреевич,
ФИО

обучающийся на 2 курсе по специальности 09.02.07 «Информационные системы и программирование» прошел учебную практику в объеме 144 часов с «6» мая 2023 г. по «2» июня 2023 г.

в организации ФГБОУ ВО «ВВГУ», Академический колледж, г. Владивосток
наименование организации, юридический адрес

В период практики в рамках осваиваемого вида профессиональной деятельности выполнял следующие виды работ:

Вид профессиональной деятельности	Код и формулировка формируемых профессиональных компетенций	Виды работ, выполненных обучающимся во время практики в рамках овладения компетенциями	Качество выполнения работ (отлично, хорошо, удовлетворительно, неудовлетворительно)
ПМ.03 Проектирование и разработка информационных систем	ПК 5.1 Собирать исходные данные для разработки проектной документации на информационную систему	Провести анализ предметной области. Обеспечивать сбор данных для анализа использования и функционирования информационной системы	<i>отлично</i>
	ПК 5.2 Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика	Разработать план мероприятий для этапов проектирования и разработки информационной системы.	<i>отлично</i>
	ПК 5.3 Разрабатывать подсистемы безопасности информационной системы в соответствии с индивидуальным заданием	Проанализировать существующие системы по выбранной тематике. Провести анализ и выборку инструментальных средств для проектирования и разработки.	<i>отлично</i>
	ПК 5.4 Производить разработку модулей информационной системы в соответствии с индивидуальным заданием	Разработать основную логику программной и интерфейсной части ИС по выбранной тематике. Разработать личный кабинет пользователя.	<i>отлично</i>
	ПК 5.5 Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы	Разработать генератор тестовых данных для тестирования работоспособности отображения информации из базы данных.	<i>отлично</i>
	ПК 5.6 Разрабатывать техническую документацию на эксплуатацию информационной системы	Оформить тестовую документацию для API интернет-магазина с подробным описанием тестовых сценариев, входных и выходных данных.	<i>отлично</i>
	ПК 5.7 Производить оценку информационной системы для выявления возможности ее модернизации	Проводить оценку качества и эффективности информационной системы. Применять методики тестирования разрабатываемой информационной системы. Формировать отчетную документацию по результатам работ. Использовать стандарты при оформлении программной документации.	<i>отлично</i>
Итоговая оценка по ПМ.03 Проектирование и разработка информационных систем			<i>отлично</i>

Вид профессиональной деятельности	Код и формулировка формируемых профессиональных компетенций	Виды работ, выполненных обучающимся во время практики в рамках овладения компетенциями	Качество выполнения работ (отлично, хорошо, удовлетворительно, неудовлетворительно)
ПМ.05 Сoadминистрирование и автоматизация баз данных и серверов	ПК 7.1 Выявлять технические проблемы, возникающие в процессе эксплуатации баз данных и серверов	Провести тестирование программного интерфейса приложения с помощью Postman.	отлично
	ПК 7.2 Осуществлять администрирование отдельных компонентов серверов	Разработать и развернуть базу данных на хостинге и интегрировать её в серверную часть информационной системы.	отлично
	ПК 7.3 Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов	Формировать необходимые для работы информационной системы требования к конфигурации локальных компьютерных сетей.	отлично
	ПК 7.4 Осуществлять администрирование баз данных в рамках своей компетенции	Настроить резервное копирование данных для базы данных информационной системы.	отлично
	ПК 7.5 Проводить аудит систем безопасности баз данных и серверов, с использованием регламентов по защите информации	Обеспечить базовую безопасность базы данных.	отлично
Итоговая оценка по ПМ.05 Сoadминистрирование и автоматизация баз данных и серверов			отлично

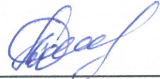
Заключение об уровне освоения обучающимися профессиональных компетенций:

освоены на продвинутом уровне.
(освоены на продвинутом уровне / освоены на базовом уровне /
освоены на пороговом уровне / освоены на уровне ниже порогового)

Дата 2 июня 2023 г.

Оценка за практику отлично

Руководитель практики от предприятия


подпись

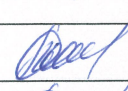
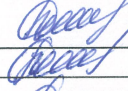
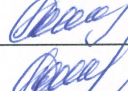
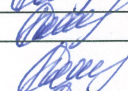
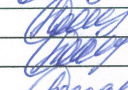
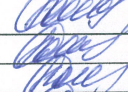
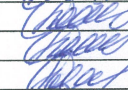
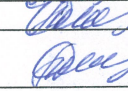
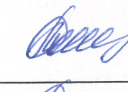
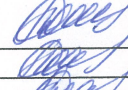
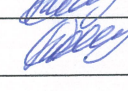
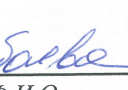
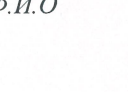
Атабаева Д. А.
Ф.И.О.

М.П.

ДНЕВНИК

прохождения учебной практики

Студент (ка) Федянин Максим Андреевич
Фамилия Имя Отчество
 Специальность 09.02.07 «Информационные системы и программирование»
 Группа С-ИП-21-1
 Место прохождения практики ФГБОУ ВО "ВВГУ", Академический колледж, г. Владивосток
 Сроки прохождения с 06.05.2023 г. по 02.06.2023 г.
 Инструктаж на рабочем месте «06» мая 2023 г Атабаева Д.А.
дата подпись Ф.И.О. инструктирующего

Дата (период)	Описание выполнения производственных заданий (виды и объем работ, выполненных за день)	Оценка	Подпись руководителя практики
06.05.2023	Прохождение инструктажа по охране труда и технике безопасности, распределение по рабочим местам, установочная лекция. Анализ деятельности предприятия ООО НПФ «ЭФФЕКТОВЕНТ». Разработка плана мероприятий для этапов проектирования и разработки интернет-магазина.	5	
08.05.2023	Анализ существующих интернет-магазинов по выбранной тематике	5	
09.05.2023	Анализ и выборка инструментальных средств для проектирования и разработки.	5	
10.05.2023	Создание карты навигации для интернет-магазина.	5	
11.05.2023	Проектирование диаграммы вариантов использования и диаграммы последовательности для интернет-магазина.	5	
12.05.2023	Проектирование диаграммы классов и макетов страниц для интернет-магазина.	5	
13.05.2023	Разработка навигационной панели интернет-магазина.	5	
15.05.2023	Разработка каталога и сегментации по параметрам для интернет-магазина.	5	
16.05.2023	Разработка возможности осуществления заказа с сайта.	5	
17.05.2023	Разработка личного кабинета пользователя и корзины.	5	
18.05.2023	Разработка статистики просмотра каждого товара и количества заказов.	5	
19.05.2023	Разработка блока «Контакты».	5	
20.05.2023	Разработка проекта базы данных интернет-магазина.	5	
22.05.2023	Разработка самой базы данных интернет-магазина.	5	
23.05.2023	Развертка базы данных на хостинге.	5	
24.05.2023	Интеграция базы данных в серверную часть интернет-магазина.	5	
25.05.2023	Обеспечение базовой безопасности базы данных.	5	
26.05.2023	Настройка резервного копирования данных для базы данных интернет-магазина.	5	
27.05.2023	Разработка генератора тестовых данных для тестирования работоспособности отображения информации из базы данных на страницах интернет-магазина.	5	
29.05.2023	Оформление тестовой документации для API интернет-магазина.	5	
30.05.2023	Тестирование API интернет-магазина с помощью Postman.	5	
31.05.2023	Выпуск и установка SSL-сертификата для интернет-магазина.	5	
01.06.2023 г.– 02.06.2023 г.	Обобщение материалов практики, оформление отчета и необходимых документов по практике.	5	

Руководитель практики

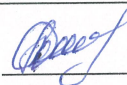
М.П.

 Атабаева Д.А.
подпись Ф.И.О.

Характеристика деятельности студента
Федянина Максима Андреевича
 группы С-ИП-21-1 при прохождении учебной практики

Код	Наименование компетенции	Основные показатели оценки результата	***Уровень (низкий, средний, высокий)
ПК 5.1	Собирать исходные данные для разработки проектной документации на информационную систему	Способность обеспечивать сбор данных для анализа использования и функционирования информационной системы. Определять состав оборудования и программных средств разработки информационной системы.	высокий
ПК 5.2	Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика	Способность разрабатывать проектную документацию на информационную систему	высокий
ПК 5.3	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием	Способность управлять процессом разработки информационных систем с использованием инструментальных средств. Модифицировать отдельные модули информационной системы.	высокий
ПК 5.4	Производить разработку модулей информационной системы в соответствии с техническим заданием	Способность разрабатывать документацию по эксплуатации информационной системы. Модифицировать отдельные модули информационной системы.	высокий
ПК 5.5	Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы	Способность применять методики тестирования разрабатываемых приложений.	высокий
ПК 5.6	Разрабатывать техническую документацию на эксплуатацию информационной системы	Способность разрабатывать проектную документацию на информационную систему. Формировать отчетную документацию по результатам работ. Использовать стандарты при оформлении программной документации.	высокий
ПК 5.7	Производить оценку информационной системы для выявления возможности ее модернизации	Способность использовать критерии оценки качества и надежности функционирования информационной системы.	высокий
ПК 7.1	Выявлять технические проблемы, возникающие в процессе эксплуатации баз данных и серверов	Способность идентифицировать технические проблемы, возникающих в процессе эксплуатации баз данных.	высокий
ПК 7.2	Осуществлять администрирование отдельных компонентов серверов	Способность участвовать в администрировании отдельных компонент серверов.	высокий
ПК 7.3	Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов	Способность формировать необходимые для работы информационной системы требования к конфигурации локальных компьютерных сетей.	высокий
ПК 7.4	Осуществлять администрирование баз данных в рамках своей компетенции	Способность участвовать в соадминистрировании серверов. Проверять наличие сертификатов на информационную систему или бизнес-приложения.	высокий
ПК 7.5	Проводить аудит систем безопасности баз данных и серверов, с использованием регламентов по защите информации	Способность проводить сертификацию информационной системы	высокий

Руководитель практики (от организации)

 Андреева Д.А.

М.П.

ХАРАКТЕРИСТИКА

о прохождении учебной практики студента

Студент Федянин Максим Андреевич С-ИП-21-1
ФИО студента *группа*
проходил практику с «06» мая 2023 г. по «02» июня 2023 г.
на базе ФГБОУ ВО "ВВГУ", Академический колледж, г. Владивосток
название предприятия
в подразделении Академический колледж
название подразделения

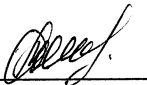
За период прохождения практики студент посетил 24 дня, из них по уважительной причине отсутствовал 0 дней, пропуски без уважительной причины составили 0 дней.

Студент соблюдал и правила техники безопасности.

За время прохождения практики студент показал, что умеет планировать и организовывать собственную деятельность, способен налаживать взаимоотношения с другими, имеет хороший уровень культуры поведения, умеет работать в команде, высокая степень сформированности умений в профессиональной деятельности.

В отношении выполнения трудовых заданий проявил такие качества, как ответственность и внимательность.

Преподаватель АК ВВГУ
Должность наставника/куратора


подпись

Атабаева Д.А.
И.О. Фамилия

М.П.