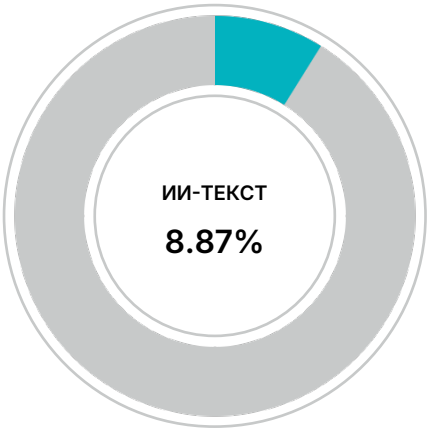
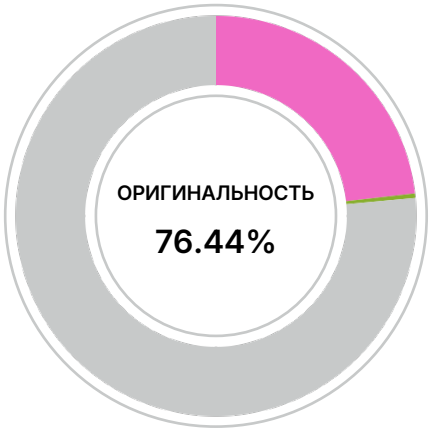
{ду}мейт
думай, пиши и проверяй

Отчёт о проверке документа

Автор: Лифанов Роман Валерьевич
Документ: Подготовка дипломного проекта (работы)
Организация: Московский университет СИНЕРГИЯ
Дата проверки: 25.04.2026, 00:31:32
Проверяющий: Пользователь API





ОРИГИНАЛЬНОСТЬ	76.44%
СОВПАДЕНИЯ	23.20%
ЦИТИРОВАНИЯ	0.36%
САМОЦИТИРОВАНИЯ	0%
ИИ-ТЕКСТ	8.87%

МАСКИРОВКИ
Не обнаружены

ИИ-ТЕКСТ Подробности по QR-code
Обнаружен ИИ-текст

СОВПАДЕНИЯ Подробности по QR-code

С Совпадение Ц Цитирование СЦ Самоцитирование

Номер	Процент	Тип	Источник	Ссылка
1	5.51%	С	МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ...	dspace.tltsu.ru
2	3.98%	С	Топ-5 ключевых составляющих архите...	synergy.ru
3	3.77%	С	Оглавление Введение 6 1 Аналитическа...	studfile.su
4	3.67%	С	РАЗРАБОТКА АВТОМАТИЗИРОВАННОГ...	lework.net
5	3.61%	С	Самарцев К.Н._ПИбд-1702в .pdf	dspace.tltsu.ru
6	2.43%	С	obrazecz-diploma-sinergii-menedzhment...	sessiusdal.ru
7	2.11%	С	ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ Р...	elibrary.ru
8	1.95%	С	НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛ...	studfile.su
9	1.61%	С	Инструменты для веб-разработки: что т...	optimalgroup.ru
10	1.52%	С	Создание темы для Wordpress.	elibrary.ru

{ду}мейт
думай, пиши и проверяй

Отчёт о проверке документа

Номер	Процент	Тип	Источник	Ссылка
11	1.48%	С	Metod_VKR_Inform.pdf	mebik.ru
12	1.48%	С	СЕРВИС ДОСТАВКИ "САМОКАТ": РАЗВ...	elibrary.ru
13	1.44%	С	Пошаговое руководство HTML: от прос...	sky.pro
14	1.4%	С	Методические рекомендации по выпол...	elibrary.ru
15	1.39%	С	Современная архитектура веб-прилож...	elibrary.ru
16	1.22%	С	<!DOCTYPE html> <html lang="de">	textur-festival.ch
17	1.19%	С	8788db9339b6c8425d370eb299219bfa....	solncesvet.ru
18	1.11%	С	Негосударственное образовательное ч...	e-biblio.ru
19	1.1%	С	Комплексный анализ производственног...	studentu24.ru
20	1.09%	С	Информационные технологии в юриспр...	biblioclub.ru
21	1.08%	С	A A A Aa Aa PDF 595404 302355 GO	ru.ruwiki.ru
22	1.04%	С	Semantic Web Accessibility Testing via Hi...	ieeexplore.ieee.org
23	1%	С	<!DOCTYPE html> <html class="no-js" la...	urait.ru
24	0.98%	С	Самокат (сервис доставки) — Википеди...	ru.wikipedia.org
25	0.95%	С	ifdocumentlocationprotocolhttp var TyntT...	tekhnosfera.com
26	0.95%	С	Т. Я. Данелян ; Международный консор...	search.rsl.ru
27	0.9%	С	Топ-5 ключевых составляющих архите...	synergy.ru
28	0.89%	С	МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ...	elib.pnzgu.ru
29	0.64%	С	Дипломная работа: Роль интеллекта в п...	homework.ru
30	0.57%	С	Топ-10 технологий веб-разработки: с че...	sky.pro
31	0.55%	С	The Expert System for Generating Front-E...	ieeexplore.ieee.org
32	0.49%	С	Действительно полезное приложение д...	pcnews.ru
33	0.44%	С	Развитие учёта, контроля и анализа в у...	book.ru
34	0.41%	Ц	Цитирование	
35	0.33%	С	Общие требования к оформлению текст...	elibrary.ru
36	0.32%	С	9 месяцев - полет нормальный	pcnews.ru
37	0.32%	С	Дневник охотника за ошибками. Путеш...	
38	0.32%	С	Radio Frequency IoT Sensors in Military O...	ieeexplore.ieee.org
39	0.32%	С	Перенаправление данных из COM-порт...	habrahabr.ru
40	0.29%	С	ПРИМЕНЕНИЕ ШАБЛОНА ПРОЕКТИРОВ...	elibrary.ru
41	0.28%	С	Влияние экономики на экологическое ...	ivo.garant.ru

{ду}мейт
думай, пиши и проверяй

Отчёт о проверке документа

Номер	Процент	Тип	Источник	Ссылка
42	0.28%	C	2024 ECOMHUB ECommerce 30 2025 In...	ecomhub.ru
43	0.28%	C	Методические рекомендации по выпол...	elibrary.ru
44	0.27%	C	Итоговая аттестация по направлению п...	biblioclub.ru
45	0.26%	C	Аутсорсинг в сфере бухгалтерского уче...	biblioclub.ru
46	0.25%	C	Разработка пользовательского интерфе...	texterra.ru
47	0.25%	C	Проектирование сайта компании пищев...	biblioclub.ru
48	0.24%	C	Blockchain Based Aadhaar	ieeexplore.ieee.org
49	0.24%	C	Упаковка как ключевой инструмент ма...	elibrary.ru
50	0.24%	C	Прекращение трудовых договоров в сл...	ivo.garant.ru
51	0.24%	C	Практическое пособие для аспирантов ...	emll.ru
52	0.23%	C	Направления совершенствования мето...	ivo.garant.ru
53	0.22%	C	Как расширить возможности Kubernete...	advis.ru
54	0.21%	C	Современные проблемы организации ...	ivo.garant.ru
55	0.19%	C	Международное санкционное право и а...	ivo.garant.ru
56	0.18%	C	Node.js. Путеводитель по технологии	
57	0.18%	C	СОЗДАНИЕ ДИЗАЙНА ОКНА РАСШИРЕ...	elibrary.ru
58	0.17%	C	Совершенствование процесса выпарив...	diss.natlib.uz
59	0.17%	C	Алтухова Н.Ф., Дзюбенко А.Л., Лосева ...	ivo.garant.ru
60	0.16%	C	CLIENT SIDE PAGE PROCESSING - ALIBA...	freepatentsonline.com
61	0.15%	C	Финансы учреждений здравоохранения...	biblioclub.ru
62	0.14%	C	Повышение быстродействия и статисти...	elib.nlb.by
63	0.13%	C	Разработка адаптивного сайта с исполь...	biblioclub.ru
64	0.12%	C	Обобщение алгоритмов математическо...	elib.nlb.by
65	0.11%	C	Актуальные проблемы и перспективы р...	elibrary.ru
66	0.11%	C	Introduction to simulation using JavaScript	ieeexplore.ieee.org
67	0.1%	C	СОВРЕМЕННЫЕ МЕТОДЫ И ИНСТРУМЕ...	elibrary.ru

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»

Факультет Информационных технологий

Специальность 09.02.07 Информационные системы и программирование *(код, наименование)* **Кафедра** ЦЭ *(аббревиатура)*

Специализация: Веб-разработка

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА – ДИПЛОМНАЯ РАБОТА (ПРОЕКТ)

на тему **РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОНЛАЙН-БРОНИРОВАНИЯ БИЛЕТОВ НА ПРИМЕРЕ ООО "САМОКАТ"»**

Обучающийся *Лифанов Роман Валерьевич* (Фамилия, Имя, Отчество) (подпись)

Руководитель Мутраков Олег Серегеевич (Фамилия, Имя, Отчество) (подпись) 6

МОСКВА 2026 г.

Автономная некоммерческая организация
высшего образования
«Московский университет «Синергия»

УТВЕРЖДАЮ
Декан факультета
Информационных технологий
Университета «Синергия»
Захаров А.В.
И. О. Фамилия

Факультет 17 Информационных технологий

Кафедра Цифровой экономики

ЗАДАНИЕ
на выпускную квалификационную работу – дипломную работу (проект)
обучающемуся
Лифанову Роману Валерьевичу

1. Тема выпускной квалификационной работы:
«Разработка веб-приложения для онлайн-бронирования билетов на примере ООО "Самокат"» 34

2. Структура ВКР.

Введение

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ 19 **РАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ**

1.1 Основные понятия и терминология в области веб-разработки
1.2 Структурные компоненты и серверные технологии современных веб-приложений
1.3 Архитектурные подходы к проектированию веб-приложений
1.4 Методологии разработки веб-приложений и их сравнительный анализ
1.5 Инструментальные средства и технологии разработки веб-приложений
1.6 Обеспечение качества и безопасности веб-приложений
Выводы по главе 1

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОНЛАЙН-БРОНИРОВАНИЯ 2 **ЛЕТОВ НА ПРИМЕРЕ ООО "САМОКАТ"**

2.1 Анализ деятельности ООО "Самокат"
2.2 Анализ текущего состояния информационной системы
2.3 Разработка веб-приложения
2.4 Функционально-технологический анализ разработанного веб-приложения
Выводы по главе 2

ГЛАВА 3. ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

3.1 Выбор и обоснование методики расчёта экономической эффективности
3.2 Расчёт показателей экономической эффективности проекта
Выводы по главе 3

Заключение

Список используемой литературы 1

Приложения

3. Основные вопросы, подлежащие разработке.

Во введении необходимо определить: актуальность темы; цель и задачи исследования; предмет и объект исследования (указывается конкретная организация и исследуемое направление ее работы); информационную базу исследования; методы исследования; практическое использование разработок и рекомендаций в деятельности исследуемой организации; структуру работы.

В главе 1 целесообразно рассмотреть: 18 овные понятия, 14 анные с разработкой веб-приложений; описать структуру и основные компоненты веб-приложений, серверные технологии; раскрыть принципы функционирования web-страниц и их взаимодействия с пользователем; изучить различные подходы к проектированию архитектуры web-приложений и веб-приложений, 2 ключая модели клиент-сервер, 15 ехуровневую архитектуру, микросервисную архитектуру, рассмотрев основные принципы, преимущества и недостатки каждого подхода; проанализировать наиболее распространенные методологии разработки веб-приложений, такие как каскадная модель, гибкие методологии (Agile, Scrum), а также их особенности, преимущества и недостатки; ознакомиться с основными инструментами и технологиями, используемыми в процессе разработки веб-приложения, включая интегрированные среды разработки (IDE), системы управления версиями, фреймворки и библиотеки, рассмотреть их функциональные возможности и область применения; уделить внимание вопросам обеспечения качества веб-приложений, включая тестирование, оптимизацию производительности, обеспечение безопасности и доступности.

В главе 2 целесообразно: рассмотреть исследуемую проблему и провести: 1) провести подробный анализ деятельности компании ООО "Самокат"; 2) проанализировать текущее состояние информационной системы компании, в том числе наличие и функциональность существующего веб-приложения; 3) описать процесс разработки веб-приложения.

В данной главе ***(разделе 2.1 и 2.2.)*** требуется рассмотреть организационную структуру, основные направления деятельности, целевую аудиторию и конкурентную среду исследуемой организации с целью определения ключевых информационных потребностей, которым должно удовлетворять разрабатываемое веб-приложение. Необходимо проанализировать текущее состояние информационной системы компании, в том числе наличие и функциональность существующего веб-приложения, выявить основные недостатки и ограничения действующего веб-приложения, а также сформулировать требования к новому с учетом потребностей бизнеса и целевой аудитории.

В разделе 2.3. следует обосновать выбор технологического стека, наиболее подходящего для реализации поставленных задач; подробно описать процесс разработки веб-приложения организации, приводя примеры технологических решений и уделить внимание вопросам обеспечения безопасности и надежности разрабатываемого веб-приложения.

В разделе 2.4. необходимо провести всесторонний анализ созданного веб-приложения по ряду ключевых критериев.

1) Необходимо оценить общую структуру и навигацию веб-приложения. 2 то подразумевает анализ логичности и интуитивной понятности расположения основных разделов, эффективности средств навигации (меню, карта, гиперссылки и т.д.) для обеспечения удобного и быстрого доступа пользователей к необходимой информации. Кроме того, следует

проанализировать визуальное оформление веб-приложения, включая дизайн, цветовую гамму, шрифты и их читаемость, а также общую эстетическую привлекательность.

2) Необходимо провести оценку информационного наполнения веб-приложения. Здесь следует проанализировать актуальность, достоверность и полноту представленной информации, ее структурированность и логичность изложения, а также удобство поиска и доступа к необходимым данным.

3) Следует уделить внимание функциональным возможностям веб-приложения. Необходимо проанализировать наличие и эффективность реализации ключевых функций, таких как обратная связь с пользователями, интеграция с социальными сетями (возможно онлайн-заказы, личный кабинет) и т.д. Также важно оценить скорость загрузки страниц и общую производительность веб-приложения.

4) Следует провести анализ соответствия веб-приложения современным стандартам и требованиям в области web-разработки. 15 включает в себя оценку кроссбраузерности и адаптивности дизайна, валидность HTML/CSS-кода, соблюдение принципов доступности для людей с ограниченными возможностями, а также оптимизацию для поисковых систем.

В главе 3 необходимо описать преимущества и экономические выгоды от внедрения нового (обновленного) веб-приложения, включая сокращение временных и трудовых затрат, улучшение качества и скорости обработки данных. Также следует рассмотреть потенциальное увеличение доходов за счет оптимизации процессов, снижение затрат на обучение сотрудников благодаря интуитивно понятному интерфейсу. Важно подкрепить выводы конкретными расчетами и сравнительным анализом до и после внедрения веб-приложения.

В заключении: обобщить основные результаты проведенного исследования. Необходимо сформулировать ключевые выводы, касающиеся актуальности и практической значимости разработанного веб-приложения 9 я деятельности предприятия, описать его функциональные возможности и преимущества, оценить эффективность внедрения с точки зрения повышения информационной прозрачности, оптимизации бизнес-процессов и расширения клиентской базы. Также следует отметить проблемы и ограничения, возникшие в ходе реализации проекта, и наметить перспективные направления дальнейшего совершенствования веб-приложения. Заключение должно логически завершать работу и подчеркивать достижение поставленной цели исследования.

Приложения.

Привести, базовую и информационно-вспомогательную информацию, используемую при написании ВКР.

4. Исходные данные по ВКР.

Основная литература: 6

1. Алексеев А. П. Введение в Web-дизайн. Учебное пособие. 8 М.: Солон-Пресс, 2021. — 184 с.

2. Тузовский А. Ф. Проектирование и разработка web-приложений. — М.: Юрайт, 2023. — 220 с.

3. Полуэктова Н. Р. Разработка веб-приложений. — М.: Юрайт, 2024. — 205 с.

Дополнительная литература:

1. Никулин В. В. Разработка серверной части веб-ресурса. Учебное пособие для вузов. — М.: Лань, 2023. — 132 с.

2. Диков А. В. Клиентские технологии веб-программирования: JavaScript и DOM. — М.: Лань, 2020. — 124 с.

3. Татро К., Макинтайр П. Создаем динамические веб-сайты на PHP. — СПб.: Питер, 2021. — 544 с.

Интернет-ресурсы:

№ п/п	Наименование Интернет-ресурса	Ссылка
1.	Тренировочная площадка для веб-разработчиков	https://www.freecodecamp.org/
2.	Архив документации	https://devdocs.io/
3.	Тестирование веб-приложения на всех популярных браузерах	https://www.lambdatest.com/
4.	Полезные инструменты для веб-разработчиков	https://smalldev.tools/
5.	Набор инструментов, помогающий дизайнеру сформировать палитру цветов.	https://colorhunt.co/

Руководитель Мутраков О.С.

подпись расшифровка

Обучающийся задание получил: «25» марта 2026 г.

Обучающийся Лифанов Р.В.

подпись расшифровка 6

СОДЕРЖАНИЕ

ВВЕДЕНИЕ……..…7
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ 6 АСПЕКТЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ.....9
1.1 Основные понятия и терминология в области веб-разработки…… 9
1.2 Структурные компоненты и серверные технологии современных веб-приложений……...11
1.3 Архитектурные подходы к проектированию веб-приложений……13
1.4 Методологии разработки веб-приложений и их сравнительный анализ... 16
1.5 Инструментальные средства и технологии разработки веб-приложений.....17
1.6 Обеспечение качества и безопасности веб-приложений.....19
ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОНЛАЙН-БРОНИРОВАНИЯ 2 ЛЕТОВ НА ПРИМЕРЕ ООО "САМОКАТ"...22
2.1 Анализ деятельности ООО "Самокат"..... 22
2.2 Анализ текущего состояния информационной системы..... 23
2.3 Разработка веб-приложения..... 25
2.4 Функционально-технологический анализ разработанного веб-приложения 51
III ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА 52
3.1. Выбор и обоснование методики расчёта экономической эффективности.....52
3.2. Расчёт показателей экономической эффективности проекта..... 53
ЗАКЛЮЧЕНИЕ.....59
СПИСОК ИСПОЛЬЗОВАННЫХ 8 ТОЧНИКОВ.....60

ВВЕДЕНИЕ

Экономическая среда современных российских предприятий характеризуется высокой конкуренцией и стремительными изменениями рыночных условий, что требует от организаций оперативного принятия управленческих решений на основе достоверной и актуальной информации. Эффективность управления предприятием напрямую зависит от качества организации доставки товаров и скорости обработки информационных потоков.

Внедрение автоматизированной «…веб-системы для доставки позволит решить указанные проблемы и создать эффективный инструмент бронирования билетов, обеспечивающий руководство компании возможность автоматизации»[34] 34

В качестве объекта исследования рассматривается ООО «Атлант»и его документооборот.

В качестве предмета исследования рассматривается функционально-технологические и организационные особенности автоматизации документооборота в ООО «Атлант».

Целью выпускной квалификационной работы является автоматизация документооборота в компании.

Задачи, которые ставятся по данному исследованию:

- Исследовать предприятие и его деятельность.
- Изучить аспекты разработки веб-приложений
- Спроектировать функциональную модель данного управленческого учёта.
- Рассмотреть примеры технологий, по которым можно будет осуществить реализацию программного обеспечения
- Спроектировать и реализовать БД по управленческому учёту.
- Произвести описание архитектуры системы и её проектирование.
- Реализовать программное обеспечение по управленческому учёту.
- Описать состав программных модулей системы.
- Разработать контрольный пример управленческого учёта.

- Обосновать экономическую эффективность проекта.

Информационная база исследования: внутренние документы и материалы ООО «Самокат»(регламенты, формы договоров, счетов, актов, первичные документы, отчётность и сведения о текущем документообороте)? нормативно-правовые и методические источники по теме.

Методы исследования: системный анализ; анализ и синтез; метод научной абстракции; экономический анализ; структурно-функциональный анализ; метод моделирования бизнес-процессов, метод функционального моделирования; метод сравнения и классификации, метод формализации и алгоритмизации, методы проектирования информационных систем, методы логического и графического анализа и другие методы.

Проектирование решения поможет существенно сократить время на составление отчётов по управлению и также позволит оперативно собирать и интерпретировать определённые сведения по учёту.

Структура работы. 61 пускная квалификационная работа состоит из введения, трёх глав, заключения, списка использованных источников и приложений.

В первой главе проводится изучение теоретических аспектов 17 работки веб-приложений.

Во второй главе представлены проектирование и реализация веб-приложений.

В третьей главе проводится расчёт экономической эффективности внедрения разработанной системы документооборота, оцениваются затраты на разработку и эксплуатацию, а также экономический эффект от автоматизации.

В заключении подводятся итоги проведённого исследования, формулируются основные выводы по результатам выполнения поставленных задач и оценивается степень достижения цели выпускной квалификационной работы.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ

1.1 Основные понятия и терминология в области веб-разработки

Веб-разработка — это процесс создания и поддержки сайтов и веб-приложений, который включает проектирование, программирование, вёрстку, настройку серверов, тестирование и дальнейшую поддержку. Для эффективной работы в этой сфере важно понимать ключевые термины и понятия.

Основные направления разработки

- Фронтенд (Frontend) — часть сайта, с которой взаимодействует пользователь: интерфейс, дизайн, интерактивные элементы (кнопки, меню, формы). Для фронтенд-разработки обычно используют HTML, CSS и JavaScript.
- Бэкенд (Backend) — серверная часть сайта, которая обрабатывает данные, выполняет логику приложения, управляет взаимодействием с базами данных, авторизацией пользователей и другими процессами «за кулисами». Для бэкенд-разработки применяют языки программирования (Python, PHP, Java, Ruby и др.) и фреймворки.
- Фулстек (Full-Stack) — разработка как клиентской (фронтенд), так и серверной (бэкенд) частей приложения.

Ключевые технологии и языки

- HTML (HyperText Markup Language) — язык разметки для создания структуры веб-страниц. Определяет заголовки, абзацы, ссылки, изображения и другие элементы.
- CSS (Cascading Style Sheets) — язык стилей, который отвечает за внешний вид веб-страниц: цвета, шрифты, отступы, расположение элементов.
- JavaScript — язык программирования, который добавляет интерактивность на веб-страницы: обрабатывает действия пользователя, изменяет содержимое страницы без перезагрузки, взаимодействует с серверами.

- API (Application Programming Interface) — интерфейс, позволяющий различным программам и сервисам взаимодействовать между собой. 30
- JSON — текстовый формат для передачи данных между сервером и веб-приложением.
- WebSocket — протокол для постоянного соединения клиента с сервером в режиме реального времени.

Системы и инструменты

- CMS (Content Management System) — платформа для управления контентом, которая упрощает создание и редактирование сайтов без глубоких технических знаний. Примеры: WordPress, Joomla, Drupal.
- Git — система контроля версий, которая позволяет отслеживать изменения в коде, работать над проектом в команде и возвращаться к предыдущим версиям. 30
- GitHub — онлайн-платформа для хранения и совместной работы над проектами.
- IDE (интегрированные среды разработки) — комплексы инструментов для разработчика, включающие текстовые редакторы, средства для компиляции кода, отладки и автоматизации сборки проекта. 9
- Фреймворки — наборы готовых библиотек, инструментов, структур и шаблонов, которые упрощают и ускоряют процесс разработки. Примеры: React, Angular, Vue.js, Laravel.

Другие важные понятия

- Домен — уникальное имя сайта в интернете (например, example.com).
- Хостинг — услуга по размещению сайта на сервере с постоянным доступом в интернет.
- SEO (Search Engine Optimization) — поисковая оптимизация, комплекс мер для повышения видимости сайта в поисковых системах.
- Адаптивный дизайн (Responsive Design) — подход, при котором сайт автоматически подстраивается под различные размеры экранов и устройств.

- СУБД (система управления базами данных) — программное обеспечение для создания, управления и обслуживания баз данных.
- SSL (Secure Sockets Layer) — технология защиты данных, передаваемых между браузером и сервером.

1.2 Структурные компоненты и серверные технологии современных веб-приложений

Структурные компоненты веб-приложений

1. Клиентская часть (фронтенд). Отвечает за взаимодействие с пользователем. Реализуется в браузере с использованием HTML (язык разметки для структуры контента), CSS (для стилизации) и JavaScript (для интерактивности). Современные фреймворки (React, Angular, Vue.js) упрощают разработку клиентской части, предлагая готовые компоненты и инструменты для создания сложных интерфейсов.

2. Серверная часть (бэкенд). Обрабатывает запросы от клиента, выполняет бизнес-логику, взаимодействует с базой данных и формирует ответ.

3. Веб-сервер. Принимает HTTP-запросы от клиентов (браузеров), обрабатывает их и отправляет ответы. Может обслуживать статические ресурсы (HTML, изображения, CSS, JavaScript) или взаимодействовать с серверными приложениями для динамической генерации контента. Примеры веб-серверов: Apache HTTP-сервер, 15 nx, Microsoft IIS.

4. Сервер приложений. 15 едназначен для выполнения динамических приложений, обработки бизнес-логики и взаимодействия с базой данных. Может выполнять сложные операции, такие как обработка запросов, работа с транзакциями и управление сеансами.

5. Базы данных. Используются для хранения 15 нных, которые извлекаются при запросе пользователя и отображаются в веб-приложении. Могут быть реляционными (MySQL, PostgreSQL, Oracle) или нереляционными (NoSQL, например MongoDB, Redis).

6. API (интерфейсы прикладного программирования). Позволяют различным компонентам системы взаимодействовать друг с другом и с внешними сервисами, обеспечивая гибкость и масштабируемость. 2

Серверные технологии

- Языки программирования:
  - Python — гибкий язык с большим количеством библиотек и фреймворков (Django, Flask).
  - Java — объектно-ориентированный язык, используемый для сложных и корпоративных веб-приложений.
  - PHP — скриптовый язык, широко применяемый для серверной разработки сайтов и веб-приложений.
  - Node.js — среда выполнения JavaScript на сервере, позволяющая использовать этот язык для серверной части.
  - C# — объектно-ориентированный язык, используемый с фреймворком ASP.NET для создания корпоративных приложений.
  - Go (Golang) — язык, отличающийся высокой производительностью и простотой, подходит для высоконагруженных проектов.
- Фреймворки:
  - Django и Flask — фреймворки на Python для быстрой разработки веб-приложений.
  - Laravel и Symfony — фреймворки на PHP.
  - ASP.NET — фреймворк на C# для разработки веб-серверов. 2
- Другие технологии:
  - AJAX (Asynchronous JavaScript and XML) — технология для асинхронного обмена данными между браузером и сервером без полной перезагрузки страницы.
  - REST API — стандарт взаимодействия между фронтендом и бэкендом, использующий стандартные HTTP-методы, передачу данных в формате JSON и чёткую структуру URL.

- Сервисы фоновых задач — используются для асинхронного выполнения операций, затрачивающих много времени (например, массовая рассылка сообщений, обработка изображений).

Дополнительные аспекты

- Облачные сервисы (Amazon Web Services, Google Cloud Platform, Microsoft Azure) предоставляют инфраструктуру как услугу (IaaS) [2] упрощают масштабирование приложений.
- Контейнеризация (например, Docker) позволяет упаковывать приложения и их зависимости в контейнеры, которые могут работать на любой системе, поддерживающей Docker. [2]
- Логирование и мониторинг помогают отслеживать работу веб-приложения, выявлять и устранять ошибки. Например, ELK Stack (Elasticsearch, Logstash, Kibana) используется для сбора, хранения и визуализации логов. [2]

1.3 Архитектурные подходы к проектированию веб-приложений

Архитектурные подходы к проектированию веб-приложений определяют структуру, компоненты и их взаимодействие, влияя на масштабируемость, гибкость, сложность разработки и поддержки системы. Некоторые из ключевых подходов:

Монолитная архитектура

Приложение разрабатывается как единое целое, где все компоненты (интерфейс, бизнес-логика, база данных) тесно связаны и работают в рамках одного процесса.

Преимущества:

- простота разработки и развёртывания;
- лёгкость тестирования всей системы как единого целого;
- подходит для небольших проектов, стартапов, MVP.

Недостатки:

- сложность масштабирования — для увеличения производительности нужно масштабировать всю систему;
- медленное развитие — изменения в одной части системы могут повлиять на другие части;
- технологическая связанность — сложно использовать разные технологии для разных частей системы.

Микросервисная архитектура

Приложение разбивается на набор независимых сервисов, каждый из которых отвечает за определённую бизнес-логику.

Преимущества:

- масштабируемость — каждый сервис можно масштабировать независимо;
- гибкость — можно использовать разные технологии для разных сервисов;
- устойчивость к сбоям — сбой одного сервиса не влияет на работу всей системы.

Недостатки:

- сложность управления системой;
- затраты на настройку оркестровки (например, в Kubernetes) и мониторинга.

Клиент-серверная архитектура

Модель взаимодействия, в которой есть три ключевых участника: клиент (пользователь), сервер (обрабатывает запросы и возвращает данные) и база данных (хранилище информации).

Ключевые особенности:

- распределение нагрузки между клиентом и сервером;
- централизованное хранение данных;
- масштабируемость — возможность увеличения числа клиентов без изменения кода.

Варианты клиент-серверной архитектуры:

- двухзвенная (двухуровневая) — прямое взаимодействие клиента с сервером;
- трёхзвенная — добавляется промежуточный уровень (например, сервер приложений);
- многоуровневая — расширение трёхзвенной архитектуры дополнительными специализированными уровнями.

Архитектурные паттерны

MVC (Model-View-Controller) — паттерн, который разделяет приложение на три компонента: модель (Model) — бизнес-логика и данные, представление (View) — пользовательский интерфейс, контроллер (Controller) — управляет взаимодействием между моделью и представлением.

MVVM (Model-View-ViewModel) — расширение MVC, где вместо контроллера используется ViewModel — связующее звено между моделью и представлением. Этот паттерн улучшает модульность и облегчает тестирование.

Другие подходы

- Serverless (бессерверная архитектура). Разработчики пишут код функций, а управление серверным окружением берёт на себя облачный провайдер. Функции запускаются только при необходимости и автоматически масштабируются в зависимости от нагрузки.
- Headless-подход. Предполагает разделение фронтенда и бэкенда: бэкенд предоставляет данные через API, а фронтенд использует эти данные для отображения контента.
- Модульные монолиты. Сочетают преимущества классических монолитных приложений с гибкостью микрофронтендов. Система разделяется на автономные модули с чёткими границами, минимальными зависимостями и изоляцией данных.

1.4 Методологии разработки веб-приложений и их сравнительный анализ

Методологии разработки веб-приложений различаются по подходам к управлению проектами, этапам работы, гибкости и взаимодействию с заказчиком. Основные из них — Waterfall (каскадная модель) и Agile (гибкие методологии), которые имеют свои особенности и области применения.

Waterfall (каскадная модель)

Суть: последовательное выполнение этапов разработки без возврата к предыдущим стадиям. Переход к следующему этапу возможен только после полного завершения текущего.

Этапы: анализ требований, проектирование, реализация, тестирование, развёртывание, поддержка.

Принципы:

- чёткое документирование всех этапов;

- невозможность пропуска этапов или возврата к предыдущим стадиям;

- изменения требований требуют пересмотра технического задания и перезапуска этапа.

Преимущества: предсказуемость сроков и бюджета, чёткая документация, простота управления проектом.

Недостатки: низкая гибкость, риск того, что продукт не будет соответствовать требованиям заказчика, если они были сформулированы некорректно, и необходимость полного перезапуска при изменении условий.

Когда применять: для проектов с чёткими, заранее определёнными требованиями, фиксированными сроками и минимальным количеством изменений.

Agile

Суть: итеративная разработка с короткими циклами (обычно 1–4 недели), регулярным взаимодействием с заказчиком и адаптацией к изменениям.

Принципы:

- итеративность;

- адаптивность к изменениям;

- фокус на работающем продукте как мере прогресса;

- сотрудничество с заказчиком на протяжении всего проекта.

Фреймворки Agile:

- Scrum. Основан на работе короткими циклами (спринтами), включает планирование спринта, ежедневные стендапы, обзор результатов и ретроспективы. Ключевые роли: Product Owner (отвечает за продукт), Scrum Master (за процесс), команда разработки.

- Kanban. Фокусируется на визуализации рабочего процесса с помощью доски с колонками («К выполнению», «В работе», «Готово»). Позволяет гибко регулировать поток задач без фиксированных спринтов.

- Extreme Programming (XP). Относится к гибким методологиям, акцентирует внимание на технических практиках: парном программировании, разработке через тестирование (TDD), непрерывной интеграции, рефакторинге.

- Lean. Направлен на устранение потерь и максимизацию ценности для клиента.

1.5 Инструментальные средства и технологии разработки веб-приложений

IDE — это комплекс инструментов для разработчика, который включает текстовый редактор, средства для компиляции кода, отладку и автоматизацию сборки проекта. Некоторые популярные IDE:

- Visual Studio Code (VS Code) — лёгкий редактор кода с открытым исходным кодом, поддерживающий множество языков и расширений. Имеет встроенную интеграцию с Git, возможности автодополнения кода, форматирования и проверки синтаксиса.

- IntelliJ IDEA — IDE, разработанная в первую очередь для Java, но поддерживающая множество других языков программирования с помощью плагинов. Интегрируется с системами контроля версий, предлагает средства для разработки интерфейсов и отладки. [9]

- PyCharm — специализированная IDE для разработки на Python, поддерживающая множество фреймворков и языков.

- NetBeans — бесплатная среда разработки, подходящая для создания Java-приложений, поддерживающая различные библиотеки и компиляторы.

Языки программирования

Для бэкенд-разработки (серверной части) используются различные языки программирования. Некоторые из них:

- PHP — скриптовый язык, предназначенный для веб-разработки. 9 Позволяет создавать ресурсы с динамическим контентом, работать с формами и базами данных.

- Python — универсальный язык программирования, который упрощает создание цифровых продуктов, связанных с обработкой больших баз данных.

- Java — один из популярных языков программирования и платформ, часто используется для создания корпоративных сайтов.

- Node.js — среда, позволяющая запускать JavaScript на сервере. Используется для создания быстрых и масштабируемых бэкенд-сервисов, например API.

- Golang (Go) — процедурный и статически типизированный язык программирования с синтаксисом, похожим на C.

Фреймворки

Фреймворки — это готовые интерфейсные и серверные решения, которые помогают экономить время на написании кода с нуля. Они определяют структуру, задают правила и предоставляют набор инструментов для разработки. Примеры фреймворков:

- React — библиотека JavaScript для создания динамичных интерфейсов пользователя.

- Angular — ориентирован на создание одностраничных приложений, расширяет возможности HTML.

- Vue.js — легковесная библиотека для создания пользовательских интерфейсов. 2

Symfony, Laravel, Yii, Zend Framework — фреймворки для PHP.

- Django — веб-фреймворк на Python, удобный для создания сайтов, работающих с базами данных.

Системы управления контентом (CMS)

CMS — это системы управления цифровым продуктом, которые предоставляют готовый функционал для работы сайта. Они позволяют публиковать, обновлять, форматировать контент, исправлять ошибки и взаимодействовать с аудиторией. Примеры CMS: WordPress, Drupal, Joomla, Opencart, Magento.

Другие инструменты

- Системы контроля версий (например, Git) — отслеживают изменения в коде, облегчают обмен между участниками разработки.

- Инструменты для тестирования (Postman, Selenium) — используются для автоматизированной проверки API, тестирования веб-приложений на разных сценариях взаимодействия с пользователем.

- Встроенные в браузер инструменты разработчика (DevTools) — помогают анализировать внутренний код сайта, проверять вёрстку, запросы и производительность.

- CSS-фреймворки (Bootstrap, Tailwind CSS) — упрощают создание адаптивных веб-сайтов за счёт готовых шаблонов и стилей.

1.6 Обеспечение качества и безопасности веб-приложений

Качество веб-приложений включает соответствие функциональным требованиям, производительность, удобство использования, совместимость с различными устройствами и средами. Для обеспечения качества применяются различные виды тестирования:

- Функциональное тестирование проверяет соответствие приложения заявленным требованиям.

- Юзабилити-тестирование оценивает простоту использования приложения, навигацию, интерфейс.

- Тестирование производительности выявляет узкие места при интенсивной нагрузке, измеряет время отклика, пропускную способность и использование ресурсов.

- Тестирование совместимости проверяет работу приложения в разных браузерах, операционных системах и на различных устройствах.

- Регрессионное тестирование проводится после обновлений, чтобы убедиться, что новые изменения не нарушили существующую функциональность.

Некоторые методы и инструменты тестирования:

- SAST (Static Application Security Testing) — статический анализ исходного кода на уязвимости без его выполнения.

- DAST (Dynamic Application Security Testing) — динамический анализ, 53 имитирующий атаки на приложение в рабочей среде.

- Open Source Analysis (OSA) — исследование безопасности компонентов с открытым исходным кодом.

- Инструменты для тестирования производительности (например, Apache JMeter, LoadRunner).

Тестирование следует начинать на ранних стадиях разработки, чтобы превентивно выявлять и устранять дефекты.

Безопасность веб-приложений

Ключевые аспекты безопасности:

- Защита данных. Использование HTTPS для шифрования трафика между сервером и клиентом, а также шифрование данных «на покое» (в базах данных, на сервере приложений).

- Аутентификация и авторизация. Внедрение многофакторной аутентификации (MFA), использование протоколов OAuth 2.0, OpenID Connect, SAML, JWT для управления доступом.

- Защита от распространённых уязвимостей. Например, предотвращение SQL-инъекций (использование параметризованных запросов), межсайтового скриптинга (XSS), CSRF-атак.

- Управление доступом. Реализация принципа наименьших привилегий, контроль доступа по ролям (RBAC), обязательная двухфакторная аутентификация для административных интерфейсов.

- Регулярный аудит и мониторинг. Анализ активности приложения и системы для выявления потенциальных угроз, своевременное обнаружение вредоносной активности.

- Использование средств защиты. Например, Web Application Firewall (WAF) для анализа трафика и блокировки подозрительных запросов, системы для защиты баз данных (DAM, DBF).

Дополнительные меры:

- Регулярное обновление ПО. Своевременное обновление программного обеспечения и компонентов защиты.

- Пентестинг (пентестинг). Контролируемая попытка взлома системы для выявления слабых мест.

- Разработка политик безопасности. Внутренние документы, регулирующие технические и организационные аспекты обработки данных, доступа к системам и реагирования на инциденты.

- Обучение персонала. Повышение осведомлённости сотрудников о мерах безопасности.

Вывод: по данной главе было произведено изучение теории по созданию веб-приложений. Вся теория была доступно приведена и описана.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ОНЛАЙН-БРОНИРОВАНИЯ БИЛЕТОВ НА ПРИМЕРЕ ООО "САМОКАТ"

2.1 Анализ деятельности ООО "Самокат"

Основные характеристики

По данным на май 2024 года, «Самокат» управлял более чем 2000 даркторами и работал в 127 городах России. Компания предлагает широкий ассортимент товаров: бакалея, хлеб, овощи, мясо, снеки, бытовая химия, товары для дома, детей и животных. 12 2021 года «Самокат» предлагает расширенный ассортимент большого склада-хаба (от 1000 м²) с доставкой через даркстор. В феврале 2022 года компания открыла онлайн-витрину «Самокат Бьюти» с более чем 7 тысячами SKU косметических товаров российских и иностранных производителей.

Технологическая платформа «Самоката» автоматизирует работу на уровне даркстора, доставки, помогает в прогнозировании объёма заказов. Например, программа отслеживает товарные остатки и сроки годности в даркторах и помогает товароведам комплектовать заказы, алгоритмы рассчитывают количество заказов на неделю вперёд в каждом районе присутствия с учётом поведения пользователей и внешних факторов (погоды или дорожных работ), система контроля курьеров помогает управлять распределённой сетью доставки. 12

Финансовые показатели

По итогам 2024 года оборот «Самоката» вырос год к году на 50,5% и достиг 248,9 млрд рублей. Среднее количество выполненных заказов в день увеличилось на 30,7% до 711 тысяч. Средний чек составил 956 рублей, показав динамику роста на 14,8%.

По итогам 2025 года оборот компании вырос на 12,3% и достиг 279,3 млрд рублей. 12 IV квартале оборот «Самоката» достиг исторического

максимума — 77,9 млрд рублей. Рост составил 17,7% год к году. Среднесуточный объём заказов за квартал превысил 808,5 тыс..

Стратегия развития

В 2025 году компания перешла к новой модели — с упором на «ритейл впечатлений». Это формат, который предполагает, что клиент получает не просто товары, а полноценный пользовательский опыт. Генеральный директор «Самоката» Сергей Попов отметил, что сохранять лидерство сервису удаётся за счёт работы команды и партнёров, развития операционной модели и внимания к обратной связи пользователей. logistics.ru +1

В 2025 году сервис расширил ассортимент за счёт партнёрства с «Мегамаркетом». В приложении стали доступны десятки тысяч non-food товаров (промышленные непродовольственные товары — например, бытовая техника) с доставкой от 15 минут или по клику — от бытовой техники и обуви до люксовой косметики и парфюмерии. Сервис продолжил развивать собственную торговую марку и выпустил 965 лимитированных новинок, включая тематические линейки и эксклюзивные капсулы мерча.

«Самокат» тестирует аэродоставку в Нижнем Новгороде. Отдельным направлением стало усиление пищевой безопасности: компания внедрила цифровую проверку партнёров с формированием риск-профиля каждого производителя и усилила контроль до начала поставок. Эти подходы стали основой для отраслевого регламента доставки пищевой продукции, который разрабатывается членами Совета Федерации, общественными организациями и самим онлайн-ритейлером.

2.2 Анализ текущего состояния информационной системы

Информационная система «Самоката» представляет собой комплексную технологическую платформу, которая автоматизирует ключевые бизнес-процессы компании, включая управление даркторами, логистику, прогнозирование спроса и взаимодействие с клиентами. Система построена на

распределённой архитектуре и использует современные IT-решения для обеспечения высокой скорости обработки данных и оперативности работы. ru.wikipedia.org* +2

Основные компоненты системы

- Система управления даркстором. Отвечает за расчёт маршрутов курьеров, управление запасами, контроль сроков годности товаров и координацию работы сотрудников (товароведов, сборщиков). Программа отслеживает товарные остатки в реальном времени, помогает комплектовать заказы и оптимизировать распределение товаров между складами. trends.rbc.ru +1

- Система работы с курьерами («Мой Самокат»). Регулирует количество курьеров в реальном времени, управляет их сменами, взаимодействием с компанией. Система анализирует исторические данные о работе курьеров и распределяет предложения о работе с учётом их локации и предпочтений.

- Логистические системы. Управляют расписанием транспорта, формируют заказы, прогнозируют последствия технических сбоев. Например, при поломке транспортного средства система перераспределяет товары, чтобы минимизировать задержки.

- Система управления ассортиментом (Матрица). Анализирует спрос пользователей и на основе этих данных формирует ассортимент, привлекает новых поставщиков. Также позволяет сегментировать инфраструктуру: определять, какие склады нуждаются в широком ассортименте, а какие — в смещённом (например, с акцентом на закуски).

- Витрины. Системы, которые определяют, какие товары показывать конкретному пользователю, в какой последовательности. Их функционал схож с инструментами классических интернет-магазинов, но адаптирован под специфику сервиса доставки.

- Поддержка. Группа систем, обеспечивающих взаимодействие с клиентами, обработку запросов и оперативное решение проблем.

- Система «Луч». Внедрена в 2025 году для усиления контроля за безопасностью готовой еды и продуктов питания. Система использует анализ

больших данных для аудита поставщиков, формирует профиль рисков для каждого производителя на основе данных из проверенных цифровых источников. Позволяет в реальном времени выявлять потенциальные риски и оперативно корректировать процессы.

2.3 Разработка веб-приложения

Представим создание кода для веб-приложения:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Веб-приложение для онлайн-бронирования билетов (ООО "Самокат")
Полный CRUD для бронирований: добавление, просмотр, редактирование, отмена
"""

import http.server
import socketserver
import json
import sqlite3
import urllib.parse
from datetime import datetime, timedelta
import hashlib

PORT = 8080
DB_NAME = 'samokat.db'

# ========== HTML + CSS + JavaScript ==========
HTML_TEMPLATE = '''<!DOCTYPE 48 l>
<html lang="ru">
<head> 13
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Самокат — онлайн бронирование билетов</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif; 13
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); min-height: 100vh; padding: 20px; }
.container { max-width: 1200px; margin: 0 auto; background: white; border-radius: 20px; box-shadow: 0 20px 40px rgba(0,0,0,0.1); overflow: hidden; }
.header { background: #2c3e50; color: white; padding: 20px 30px; text-align: center; }
.header h1 { font-size: 28px; margin-bottom: 5px; }
.header p { opacity: 0.9; font-size: 14px; }
.content { padding: 30px; }
.tabs { display: flex; gap: 10px; margin-bottom: 30px; border-bottom: 2px solid #e0e0e0; flex-wrap: wrap; }
.tab { padding: 10px 25px; cursor: pointer; background: none; border: none; font-size: 16px; font-weight: 600; color: #7f8c8d; transition: 0.3s; }
.tab.active { color: #2c3e50; border-bottom: 3px solid #3498db; }
.tab-pane { display: none; animation: fadeIn 0.3s; }
.tab-pane.active { display: block; }
@keyframes fadeIn { from { opacity: 0; } to { opacity: 1; } }
.form-group { margin-bottom: 20px; }
label { display: block; margin-bottom: 8px; font-weight: 13 ); color: #2c3e50; }
input, select, textarea { width: 100%; padding: 12px; border: 2px solid #e0e0e0; border-radius: 10px; font-size: 13 x; transition: 0.3s; }
input:focus, select:focus, textarea:focus { outline: none; border-color: #3498db; }
```

button { background: #3498db; color: white; padding: 12px 30px; border: none; border-radius: 10px; cursor: pointer; font-size: 16px; font-weight: 600; transition: 0.3s; margin-right: 10px; margin-top: 10px; }

button:hover { background: #2980b9; transform: translateY(-2px); }

button.danger { background: #e74c3c; }

button.danger:hover { background: #c0392b; }

button.success { background: #27ae60; }

button.success:hover { background: #229954; }

.route-card, .booking-card { background: #f8f9fa; border-radius: 15px; padding: 15px; margin-bottom: 15px; cursor: pointer; transition: 0.3s; border: 2px solid transparent; }

.route-card:hover { border-color: #3498db; background: #fff; }

.route-card.selected { border-color: #3498db; background: #e3f2fd; }

.booking-card { cursor: default; border: 1px solid #e0e0e0; }

.booking-card:hover { box-shadow: 0 5px 15px rgba(0,0,0,0.1); }

.route-title { font-weight: bold; font-size: 18px; color: #2c3e50; }

.route-details, .booking-details { display: flex; justify-content: space-between; margin-top: 10px; color: #7f8c8d; flex-wrap: wrap; gap: 10px; }

.price { color: #27ae60; font-weight: bold; font-size: 20px; }

.seats { display: grid; grid-template-columns: repeat(8, 1fr); gap: 10px; margin: 20px 0; }

.seat { background: #ecf0f1; text-align: center; padding: 10px; border-radius: 10px; cursor: pointer; transition: 0.2s; }

.seat:hover { background: #bdc3c7; }

.seat.booked { background: #e74c3c; cursor: not-allowed; opacity: 0.6; }

.seat.selected { background: #27ae60; color: white; }

.booking-info { background: #ecf0f1; padding: 20px; border-radius: 15px; margin-top: 20px; }

.success { background: #d4edda; color: #155724; padding: 15px; border-radius: 10px; margin-bottom: 20px; }

```
        .error { background: #f8d7da; color: #721c24; padding: 15px; border-radius: 10px; margin-bottom: 20px; }
        .flex { display: flex; gap: 20px; flex-wrap: wrap; }
        .flex > div { flex: 1; }
        .modal { display: none; position: fixed; top: 0; left: 0; width: 100%; height: 100%; background: rgba(0,0,0,0.5); z-index: 1000; justify-content: center; align-items: center; }
        .modal-content { background: white; padding: 30px; border-radius: 20px; max-width: 500px; width: 90%; max-height: 80vh; overflow-y: auto; }
        .modal-header { display: flex; justify-content: space-between; margin-bottom: 20px; }
        .close { cursor: pointer; font-size: 28px; font-weight: bold; color: #7f8c8d; }
        .close:hover { color: #2c3e50; }
        @media (max-width: 768px) { .seats { grid-template-columns: repeat(4, 1fr); } }
    </style>
</head>
<body>
<div class="container">
    <div class="header">
        <h1>🚌 Самокат — онлайн бронирование билетов</h1>
        <p>Удобный сервис для покупки автобусных билетов</p>
    </div>
    <div class="content">
        <div class="tabs">
            <button class="tab active" onclick="showTab('search')">🔍 Поиск билетов</button> 16
```

```
            <button class="tab" onclick="showTab('my')">📋 Мои бронирования</button>
            <button class="tab" onclick="showTab('all')">📋 Все брони (админ)</button>
        </div>

        <!-- Панель поиска и бронирования -->
        <div id="search-tab" class="tab-pane active">
            <div class="flex">
                <div>
                    <div class="form-group">
                        <label>Откуда</label>
                        <input type="text" id="from_city" placeholder="Например: Москва">
                    </div>
                </div>
                <div>
                    <div class="form-group">
                        <label>Куда</label>
                        <input type="text" id="to_city" placeholder="Например: Санкт-Петербург">
                    </div>
                </div>
                <div>
                    <div class="form-group">
                        <label>Дата поездки</label>
                        <input type="date" id="date">
                    </div>
                </div>
```

```
                <div style="display: flex; align-items: end;">
                    <button onclick="searchRoutes()">Найти билеты</button>
                </div>
            </div>
            <div id="search-results" style="margin-top: 30px;"></div>
        </div>

        <!-- Панель моих бронирований (по имени) -->
        <div id="my-tab" class="tab-pane">
            <div class="form-group">
                <label>Ваше имя для поиска бронирований:</label>
                <input type="text" id="search_name" placeholder="Введите имя пассажира">
                <button onclick="loadMyBookings()" style="margin-top: 10px;">Найти мои брони</button>
            </div>
            <div id="my-bookings"></div>
        </div>

        <!-- Панель всех бронирований -->
        <div id="all-tab" class="tab-pane">
            <h3>Все бронирования (администрирование)</h3>
            <button onclick="loadAllBookings()" style="margin-bottom: 20px;">Обновить список</button>
            <div id="all-bookings"></div>
        </div>
    </div>
</div>

<!-- Модальное окно редактирования -->
```

```
<div id="editModal" class="modal">
    <div class="modal-content">
        <div class="modal-header">
            <h3>✎ Редактирование бронирования</h3>
            <span class="close" 16 lick="closeModal()">&times;</span>
        </div>
        <div id="edit-form">
            <div class="form-group">
                <label>Имя пассажира:</label>
                <input type="text" id="edit_passenger_name">
            </div>
            <div class="form-group">
                <label>Номер места:</label>
                <input type="number" id="edit_seat_number" min="1" max="40">
            </div>
            <button onclick="saveBookingEdit()" class="success">Сохранить
изменения</button>
            <button onclick="closeModal()">Отмена</button>
        </div>
    </div> 31
</div>

<script>
    let currentRouteId = null;
    let currentSelectedSeat = null;
    let currentEditBookingId = null;

    function showTab(tab) {
```

```
            document.querySelectorAll('.tab').forEach(t => t.classList.remove('active'));
            document.querySelectorAll('.tab-pane').forEach(p => p.classList.remove('active'));
            if (tab === 'search') {
                document.querySelector('#search-tab').classList.add('active');
                document.querySelectorAll('.tab')[0].classList.add('active');
            } else if (tab === 'my') {
                document.querySelector('#my-tab').classList.add('active');
                document.querySelectorAll('.tab')[1].classList.add('active');
            } else {
                document.querySelector('#all-tab').classList.add('active');
                document.querySelectorAll('.tab')[2].classList.add('active');
                loadAllBookings();
            }
        }

        async function searchRoutes() {
            const from = document.getElementById('from_city').value;
            const to = document.getElementById('to_city').value;
            const date = document.getElementById('date').value;
            if (!from || !to) { alert('Заполните города'); return; }
            const res = await fetch(`/api/routes?from=${encodeURIComponent(from)}&to=${encodeURIComponent(to)}&date=${date}`);
            const routes = await res.json();
            const container = document.getElementById('search-results');
            if (routes.length === 0) { container.innerHTML = '<div class="error">Рейсов не найдено</div>'; return; }
```

```
        container.innerHTML = '<h3>Доступные рейсы:</h3>' + routes.map(r => `
            <div class="route-card" onclick="selectRoute(${r.id}, '${r.from_city}', '${r.to_city}', ${r.price})">
                <div class="route-title">${r.from_city} → ${r.to_city}</div>
                <div class="route-details">
                    <span>🕐 ${r.departure_time}</span>
                    <span>📅 ${r.trip_date}</span>
                    <span>💰 <span class="price">${r.price} ₽</span></span>
                    <span> Свободно: ${r.free_seats}</span>
                </div>
            </div>
        `).join('');
    }

    function selectRoute(routeId, from, to, price) {
        currentRouteId = routeId;
        document.getElementById('search-results').innerHTML = `
            <h3>Выберите место (рейс: ${from} → ${to}, ${price} ₽)</h3>
            <div id="seats-container" class="seats"></div>
            <div class="booking-info">
                <div id="selected-seat-info"></div>
                <button onclick="bookSeat()">Забронировать</button>
                <button onclick="searchRoutes()">← Назад к рейсам</button>
            </div>
        `;
        loadSeats(routeId);
    }
```

```
async function loadSeats(routeId) {
    const res = await fetch(`/api/seats?route_id=${routeId}`);
    const seats = await res.json();
    const container = document.getElementById('seats-container');
    if (!container) return;
    container.innerHTML = seats.map(s => `
        <div class="seat ${s.is_booked ? 'booked' : ''}" data-seat="${s.seat_number}" onclick="selectSeat(${s.seat_number}, ${s.is_booked})">
            ${s.seat_number}
        </div>
    `).join('');
}

function selectSeat(seatNum, isBooked) {
    if (isBooked) { alert('Место уже занято'); return; }
    document.querySelectorAll('.seat').forEach(s => s.classList.remove('selected'));
    const seatDiv = document.querySelector(`.seat[data-seat="${seatNum}"]`);
    if (seatDiv) seatDiv.classList.add('selected');
    currentSelectedSeat = seatNum;
    document.getElementById('selected-seat-info').innerHTML = `Выбрано место №${seatNum}`;
}

async function bookSeat() {
    if (!currentRouteId || !currentSelectedSeat) { alert('Выберите место'); return; }
    const passengerName = prompt('Введите ваше имя:');
```

```
        if (!passengerName || passengerName.trim() === '') { alert('Имя обязательно'); return; }
        const res = await fetch('/api/book', {
          method: 'POST',
          headers: { 'Content-Type': 'application/json' },
          body: JSON.stringify({ 36
            route_id: currentRouteId,
            seat_number: currentSelectedSeat,
            passenger_name: passengerName.trim()
          })
        });
        const data = await res.json();
        if (res.ok) {
          alert(`✅ Билет забронирован!\\nКод: ${data.booking_code}\\nМесто: ${currentSelectedSeat}\\nПассажир: ${passengerName}`);
          searchRoutes();
        } else {
          alert('❌ Ошибка: ' + data.error);
        }
      }

      async function loadMyBookings() {
        const passengerName = document.getElementById('search_name').value;
        if (!passengerName) { alert('Введите имя для поиска'); return; }
        const res = await fetch(`/api/my-bookings?name=${encodeURIComponent(passengerName)}`);
        const bookings = await res.json();
        const container = document.getElementById('my-bookings');
```

```
            if (bookings.length === 0) { container.innerHTML = '<div class="error">Бронирований для этого имени не найдено</div>'; return; }
            container.innerHTML = '<h3>Ваши бронирования:</h3>' + bookings.map(b => `
                <div class="booking-card">
                    <div class="route-title">${b.from_city} 23 {b.to_city}</div>
                    <div class="booking-details">
                        <span>📅 Дата: ${b.trip_date}</span>
                        <span>🕐 Время: ${b.departure_time}</span>
                        <span> Место: №${b.seat_number}</span>
                        <span>💰 Цена: ${b.price} ₽</span>
                        <span>🔑 Код: <b>${b.booking_code}</b></span>
                        <span>👤 Пассажир: ${b.passenger_name}</span>
                    </div>
                    <button onclick="editBooking(${b.id}, ${b.seat_number}, '${b.passenger_name}')" style="background: #f39c12;">✏️ Редактировать</button>
                    <button onclick="cancelBooking(${b.id})" class="danger">❌ Отменить</button>
                </div>
            `).join('');
        }

        async function loadAllBookings() {
            const res = await fetch('/api/all-bookings');
            const bookings = await res.json();
            const container = document.getElementById('all-bookings');
            if (bookings.length === 0) { 23 ntainer.innerHTML = '<div class="error">Нет бронирований</div>'; return; } 10
```

```
            container.innerHTML = bookings.map(b => `
                <div class="booking-card">
                    <div class="route-title">${b.from_city} → ${b.to_city}</div>
                    <div class="booking-details">
                        <span>📅 ${b.trip_date}</span>
                        <span>🕐 ${b.departure_time}</span>
                        <span>🪑 Место ${b.seat_number}</span>
                        <span>💰 ${b.price} ₽</span>
                        <span>🔑 ${b.booking_code}</span>
                        <span>👤 ${b.passenger_name}</span>
                    </div>
                    <button onclick="editBooking(${b.id}, ${b.seat_number}, '${b.passenger_name}')" style="background: #f39c12;">✏️ Редактировать</button>
                    <button onclick="cancelBooking(${b.id})" class="danger">❌ Отменить</button>
                </div>
            `).join('');
        }

        function editBooking(id, seatNum, passengerName) {
            currentEditBookingId = id;
            document.getElementById('edit_passenger_name').value = passengerName;
            document.getElementById('edit_seat_number').value = seatNum;
            document.getElementById('editModal').style.display = 'flex';
        }

        async function saveBookingEdit() {
```

```
        const newName =
document.getElementById('edit_passenger_name').value;
        const newSeat =
parseInt(document.getElementById('edit_seat_number').value);
        if (!newName || !newSeat) { alert('Заполните все поля'); return; }
        const res = await fetch('/api/update-booking', {
          method: 'POST',
          headers: { 'Content-Type': 'application/json' },
          body: JSON.stringify({ 36
            booking_id: currentEditBookingId,
            passenger_name: newName,
            seat_number: newSeat
          })
        });
        const data = await res.json();
        if (res.ok) {
          alert('✅ Бронирование обновлено');
          closeModal();
          loadAllBookings();
          const searchName = document.getElementById('search_name').value;
          if (searchName) loadMyBookings();
        } else {
          alert('❌ Ошибка: ' + data.error);
        }
      }

      async function cancelBooking(bookingId) {
        if (!confirm('Отменить бронирование?')) return;
        const res = await fetch('/api/cancel', {
```

```
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ 36king_id: bookingId })
            });
            if (res.ok) {
                alert('Бронирование отменено');
                loadAllBookings();
                const searchName = document.getElementById('search_name').value;
                if (searchName) loadMyBookings();
            } else alert('Ошибка отмены');
        }

        function closeModal() {
            document.getElementById('editModal').style.display = 'none';
            currentEditBookingId = null;
        }

        // Инициализация даты
        document.getElementById('date').value = new
Date().toISOString().slice(0,10);

        // Закрытие модального окна при клике вне его
        window.onclick = function(event) {
            const modal = document.getElementById('editModal');
            if (event.target === modal) closeModal();
        }
    </script>
    </body>
    </html>
    '''
```

```
# ========== HTTP ОБРАБОТЧИК ==========
class SamokatHTTPHandler(http.server.BaseHTTPRequestHandler):

    def do_GET(self):
        parsed = urllib.parse.urlparse(self.path) 32
        path = parsed.path

        if path == '/' or path == '/index.html':
            self._init_db()
            self.send_response(200)
            self.send_header('Content-type', 'text/html; charset=utf-8')
            self.end_headers()
            self.wfile.write(HTML_TEMPLATE.encode('utf-8')) 32

        elif path == '/api/routes':
            params = urllib.parse.parse_qs(parsed.query)
            from_city = params.get('from', [''])[0]
            to_city = params.get('to', [''])[0]
            date = params.get('date', [''])[0]
            self._send_json(self._search_routes(from_city, to_city, date))

        elif path == '/api/seats':
            params = urllib.parse.parse_qs(parsed.query)
            route_id = int(params.get('route_id', [0])[0])
            self._send_json(self._get_seats(route_id))

        elif path == '/api/my-bookings':
            params = urllib.parse.parse_qs(parsed.query)
            name = params.get('name', [''])[0]
```

```
            self._send_json(self._get_bookings_by_name(name))

        elif path == '/api/all-bookings':
            self._send_json(self._get_all_bookings())

        else:
            self.send_response(404)
            self.end_headers()

    def do_POST(self):
        content_length = int(self.headers.get('Content-Length', 0))
        post_data = self.rfile.read(content_length)
        data = json.loads(post_data.decode('utf-8'))

        if self.path == '/api/book':
            result = self._create_booking(data)
            self._send_json(result, 200 if 'booking_code' in result else 400)

        elif self.path == '/api/cancel':
            result = self._cancel_booking(data.get('booking_id'))
            self._send_json(result, 200 if result.get('success') else 400)

        elif self.path == '/api/update-booking':
            result = self._update_booking(data)
            self._send_json(result, 200 if result.get('success') else 400)

        else:
            self.send_response(404)
            self.end_headers()
```

```
def _send_json(self, data, status=200):
    self.send_response(status)
    self.send_header('Content-type', 'application/json; charset=utf-8')
    self.end_headers()
    self.wfile.write(json.dumps(data, 32 ure_ascii=False).encode('utf-8'))

# ---------- Работа с БД ----------
def _init_db(self):
    conn = sqlite3.connect(DB_NAME)
    c = conn.cursor()

    # Таблица маршрутов
    c.execute('''CREATE TABLE IF NOT EXISTS routes (
        id INTEGER PRIMARY KEY,
        from_city TEXT,
        to_city TEXT,
        departure_time TEXT,
        trip_date TEXT,
        price INTEGER,
        total_seats INTEGER DEFAULT 40
    )''')

    # Таблица бронирований
    c.execute('''CREATE TABLE IF NOT EXISTS bookings (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        route_id INTEGER,
        seat_number INTEGER,
        passenger_name TEXT,
        booking_code TEXT,
        booked_at TEXT,
```

```
            FOREIGN KEY(route_id) REFERENCES routes(id),
            UNIQUE(route_id, seat_number)
        )'''')

        # Заполнение тестовыми маршрутами
        c.execute("SELECT COUNT(*) FROM routes")
        if c.fetchone()[0] == 0:
            today = datetime.now().strftime('%Y-%m-%d')
            tomorrow = (datetime.now() + timedelta(days=1)).strftime('%Y-%m-
%d')
            day2 = (datetime.now() + timedelta(days=2)).strftime('%Y-%m-%d')

            routes_data = [
                (1, 'Москва', 'Санкт-Петербург', '08:00', today, 1500),
                (2, 'Москва', 'Санкт-Петербург', '20:00', today, 1800),
                (3, 'Москва', 'Казань', '09:30', today, 1200),
                (4, 'Санкт-Петербург', 'Москва', '10:00', tomorrow, 1500),
                (5, 'Екатеринбург', 'Челябинск', '14:00', today, 800),
                (6, 'Москва', 'Нижний Новгород', '11:00', day2, 1000),
                (7, 'Казань', 'Москва', '15:30', tomorrow, 1250),
            ]
            c.executemany("INSERT OR IGNORE INTO routes (id, from_city,
to_city, departure_time, trip_date, price) VALUES (?,?,?,?,?,?)", routes_data)

        # Тестовое бронирование для примера
        c.execute("SELECT COUNT(*) FROM bookings")
        if c.fetchone()[0] == 0:
            test_booking_code =
hashlib.md5(f"test1{datetime.now()}".encode()).hexdigest()[:8].upper()
```

```
                c.execute("INSERT INTO bookings (route_id, seat_number,
passenger_name, booking_code, booked_at) VALUES (?,?,?,?,?)",
                          (1, 5, 'Иван Петров', test_booking_code,
datetime.now().isoformat()))

            conn.commit()
            conn.close()

        def _search_routes(self, from_city, to_city, date):
            self._init_db()
            conn = sqlite3.connect(DB_NAME)
            c = conn.cursor()

            query = '''SELECT r.id, r.from_city, r.to_city, r.departure_time, r.trip_date,
r.price,
                      (r.total_seats - COUNT(b.id)) as free_seats
                      FROM routes r
                      LEFT JOIN bookings b ON r.id = b.route_id
                      WHERE r.from_city LIKE ? AND r.to_city LIKE ?
                      AND (? = '' OR r.trip_date = ?)
                      GROUP BY r.id
                      HAVING free_seats > 0'''

            like_from = '%' + from_city + '%'
            like_to = '%' + to_city + '%'
            c.execute(query, (like_from, like_to, date, date))
            rows = c.fetchall()
            conn.close()

            return [{'id': r[0], 'from_city': r[1], 'to_city': r[2],
```

```
                'departure_time': r[3], 'trip_date': r[4], 'price': r[5], 'free_seats': r[6]}
for r in rows]

        def _get_seats(self, route_id):
            conn = sqlite3.connect(DB_NAME)
            c = conn.cursor()
            c.execute("SELECT total_seats FROM routes WHERE id=?", (route_id,))
            total = c.fetchone()
            if not total:
                return []
            total_seats = total[0]
            c.execute("SELECT seat_number FROM bookings WHERE route_id=?",
(route_id,))
            booked = {row[0] for row in c.fetchall()}
            conn.close()

            seats = []
            for i in range(1, total_seats + 1):
                seats.append({'seat_number': i, 'is_booked': i in booked})
            return seats

        def _create_booking(self, data):
            route_id = data.get('route_id')
            seat_number = data.get('seat_number')
            passenger_name = data.get('passenger_name', '').strip()

            if not passenger_name:
                return {'error': 'Введите имя пассажира'}
            if not route_id or not seat_number:
                return {'error': 'Не выбран рейс или место'}
```

```
        conn = sqlite3.connect(DB_NAME)
        c = conn.cursor()

        # Проверка существования рейса
        c.execute("SELECT id FROM routes WHERE id=?", (route_id,))
        if not c.fetchone():
            conn.close()
            return {'error': 'Рейс не найден'}

        # Проверка, что место свободно
        c.execute("SELECT id FROM bookings WHERE route_id=? AND seat_number=?", (route_id, seat_number))
        if c.fetchone():
            conn.close()
            return {'error': f'Место {seat_number} уже занято'}

        # Создание бронирования
        booking_code = hashlib.md5(f"{route_id}{seat_number}{datetime.now()}{passenger_name}".encode()).hexdigest()[:8].upper()
        c.execute("INSERT INTO bookings (route_id, seat_number, passenger_name, booking_code, booked_at) VALUES (?,?,?,?,?)",
                  (route_id, seat_number, passenger_name, booking_code, datetime.now().isoformat()))
        conn.commit()
        conn.close()

        return {'booking_code': booking_code, 'success': True}
```

```
        def _get_bookings_by_name(self, name):
            if not name:
                return []
            conn = sqlite3.connect(DB_NAME)
            c = conn.cursor()
            c.execute("SELECT b.id, r.from_city, r.to_city, r.trip_date,
r.departure_time, b.seat_number, r.price, b.booking_code, b.passenger_name FROM
bookings b JOIN routes r ON b.route_id = r.id WHERE b.passenger_name LIKE ?
ORDER BY b.id DESC", ('%' + name + '%',))
            rows = c.fetchall()
            conn.close()

            return [{'id': r[0], 'from_city': r[1], 'to_city': r[2], 'trip_date': r[3],
                    'departure_time': r[4], 'seat_number': r[5], 'price': r[6],
                    'booking_code': r[7], 'passenger_name': r[8]} for r in rows]

        def _get_all_bookings(self):
            conn = sqlite3.connect(DB_NAME)
            c = conn.cursor()
            c.execute('''SELECT b.id, r.from_city, r.to_city, r.trip_date,
r.departure_time,
                            b.seat_number, r.price, b.booking_code, b.passenger_name
                        FROM bookings b
                        JOIN routes r ON b.route_id = r.id
                        ORDER BY b.id DESC''')
            rows = c.fetchall()
            conn.close()

            return [{'id': r[0], 'from_city': r[1], 'to_city': r[2], 'trip_date': r[3],
                    'departure_time': r[4], 'seat_number': r[5], 'price': r[6],
```

```
            'booking_code': r[7], 'passenger_name': r[8]} for r in rows]

    def _cancel_booking(self, booking_id):
        if not booking_id:
            return {'error': 'ID бронирования не указан'}

        conn = sqlite3.connect(DB_NAME)
        c = conn.cursor()
        c.execute("DELETE FROM bookings WHERE id=?", (booking_id,))
        conn.commit()
        affected = c.rowcount
        conn.close()

        if affected:
            return {'success': True}
        return {'error': 'Бронирование не найдено'}

    def _update_booking(self, data):
        booking_id = data.get('booking_id')
        new_name = data.get('passenger_name', '').strip()
        new_seat = data.get('seat_number')

        if not booking_id:
            return {'error': 'ID не указан'}
        if not new_name:
            return {'error': 'Имя не может быть пустым'}
        if not new_seat or new_seat < 1 or new_seat > 40:
            return {'error': 'Некорректный номер места (1-40)'}

        conn = sqlite3.connect(DB_NAME)
```

```
        c = conn.cursor()

        # Получаем route_id текущего бронирования
        c.execute("SELECT route_id, seat_number FROM bookings WHERE id=?", (booking_id,))
        result = c.fetchone()
        if not result:
            conn.close()
            return {'error': 'Бронирование не найдено'}

        route_id, old_seat = result

        # Проверяем, что новое место свободно (если оно отличается от старого)
        if new_seat != old_seat:
            c.execute("SELECT id FROM bookings WHERE route_id=? AND seat_number=?", (route_id, new_seat))
            if c.fetchone():
                conn.close()
                return {'error': f'Место {new_seat} уже занято'}

        # Обновляем
        c.execute("UPDATE bookings SET passenger_name=?, seat_number=? WHERE id=?",
                  (new_name, new_seat, booking_id))
        conn.commit()
        conn.close()

        return {'success': True}
```

```
    def log_message(self, format, *args):
        # Исправленная версия логирования
        try:
            message = format % args
        except:
            message = str(args)
        print(f"[{datetime.now().strftime('%H:%M:%S')}] {message}")

# ========== ЗАПУСК ==========
if __name__ == '__main__':
    with socketserver.TCPServer(("", PORT), SamokatHTTPHandler) as httpd:
        print("=" * 60)
        print("🚌 ООО «Самокат» — система онлайн-бронирования билетов")
        print("=" * 60)
        print(f"✅ Сервер запущен на http://localhost:{PORT}")
        print(f"🗂 База данных: {DB_NAME}")
        print("🔄 Доступные операции:")
        print("   • Поиск рейсов и бронирование")
        print("   • Просмотр бронирований по имени")
        print("   • Админ-панель со всеми бронями")
        print("   • Редактирование имени и места")
        print("   • Отмена бронирования")
        print("=" * 60)
        print("Нажмите Ctrl+C для остановки\n")
                                httpd.serve_forever()
```

2.4 Функционально-технологический анализ разработанного веб-приложения

На рисунке 1 представлена страница всех броней.

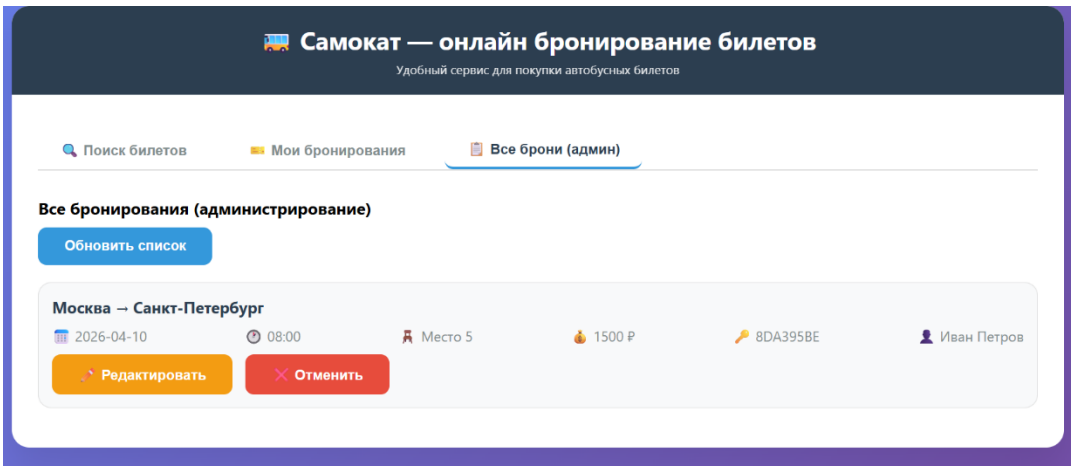


Рисунок 1 – Веб-страница всех броней

На рисунке 2 представлена страница броней клиента.

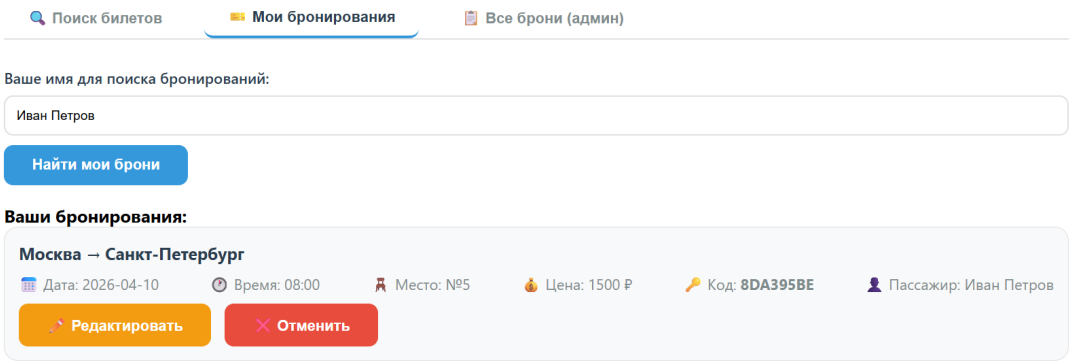


Рисунок 2 – Веб-страница броней клиента

Вывод: По данной главе было произведено изучение предприятия, и его информационной системы, которая есть на данный момент. Также было произведено создание информационной системы бронирования и проведён функционально-технологический анализ.

ГЛАВА III. ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

3.1. Выбор и обоснование методики расчёта экономической эффективности

Экономическая сущность исследуемой задачи напрямую зависит от специфики бизнес-процесса. Функциями данного бизнес-процесса в рамках реализации информационной системы:

- Сохранение истории взаимодействия с клиентом;
- Хранение и анализ данных о клиентах компании;
- Анализ работы менеджера по работе с клиентами;

Создание документации, обеспечение функционирования электронного документооборота;

Сегментирование клиентской базы;

Автоматизация процесса проведения продажи в результате взаимодействия с клиентом;

Разрабатываемый модуль, который автоматизирует бизнес-процесс взаимодействия для построения отчетности по документообороту. Результатом работы системы будут мониторинговые и аналитические отчеты.

Перед планированием хода выполнения работ над бизнес-процессом, рассмотрим жизненный цикл экономической информационной системы. Основными этапами ЖЦ являются:

- Пред проектная работа:
- Сбор информации;
- Анализ информации.
- Проектирование информационной системы.
- Разработка прототипа информационной системы.
- Тестирование.

3.2. Расчёт показателей экономической эффективности проекта

В соответствии с методикой необходимо провести расчёт показателей экономической эффективности при сравнении базового и проектного вариантов.

Для расчетов трудоёмкости базового и проектного вариантов решения задачи составлены таблицы характеристики затрат на обработку информации по базовому и проектному варианту.

В таблице 14 приведены результаты расчета затрат на обработку информации по базовому варианту.

Таблица 14

Затраты на обработку информации по базовому варианту

№ п/п	Наименование операции	Единица измерения	Объём работы в год	Норма выработки (операций в час)	Трудоёмкость, ч.
1	Входящая заявка от клиентов	Заявка	200	10	20
2	Перенаправление заявки ответственному исполнителю	Поручение	200	60	3,33
3	Определение нормативного срока решения заявки	Расчет	200	30	6,67
4	Учет выполненных работ по заявке	Запись	2500	20	125
5	Расчет трудоемкости заявки	Документ	200	20	10
6	Выборка актуальных заявок и (или) заявок, удовлетворяющих критерию поиска	Выборка	150	10	15
7	Статистический отчет по обработке заявок (с группировкой по разным признакам: количеству и трудоемкости по месяцам, исполнителям, статусам, типам)	Отчет	13	4	3,25

№ п/п	Наименование операции	Единица измерения	Объём работы в год	Норма выработки (операций в час)	Трудоёмкость, ч.
8	Отчет по учету выполнения и обработке заявок	Отчет	13	10	1,3
9	Отчет по динамике обработки заявок	Отчет	13	2	6,5
	Всего:			166,00	191,05

В таблице 15 приведены результаты расчета затрат на обработку информации по проектному варианту.

Таблица 15

Затраты на обработку информации по проектному варианту

№ п/п	Наименование операции	Единица измерения	Объём работы в год	Норма выработки (операций в час)	Трудоёмкость,
1	Входящая заявка от клиентов	Заявка	200	30	6,67
2	Перенаправление заявки ответственному исполнителю	Поручение	200	360	0,56
3	Определение нормативного срока решения заявки	Расчет	200	360	0,56
4	Учет выполненных работ по заявке	Запись	2500	120	20,83
5	Расчет стоимости заявки	Документ	200	60	3,33
6	Выборка актуальных заявок и (или) заявок, удовлетворяющих критерию поиска	Выборка	150	120	1,25
7	Статистический отчет по обработке заявок (с группировкой по разным признакам: количеству и доходам по месяцам,	Отчет	13	360	0,04

№ п/п	Наименование операции	Единица измерения	Объём работы в год	Норма выработки (операций в час)	Трудоёмкость, 1
	исполнителям, статусам, типам)				
8	Отчет по учету выполнения и обработке заявок	Отчет	13	240	0,05
9	Отчет по динамике обработки заявок	Отчет	13	240	0,05
Всего:				1415	33,34

Рассчитанные показатели трудоёмкости базового и проектного вариантов используются для вычисления показателей абсолютных и относительных изменений затрат, а также индекса изменения затрат. 1

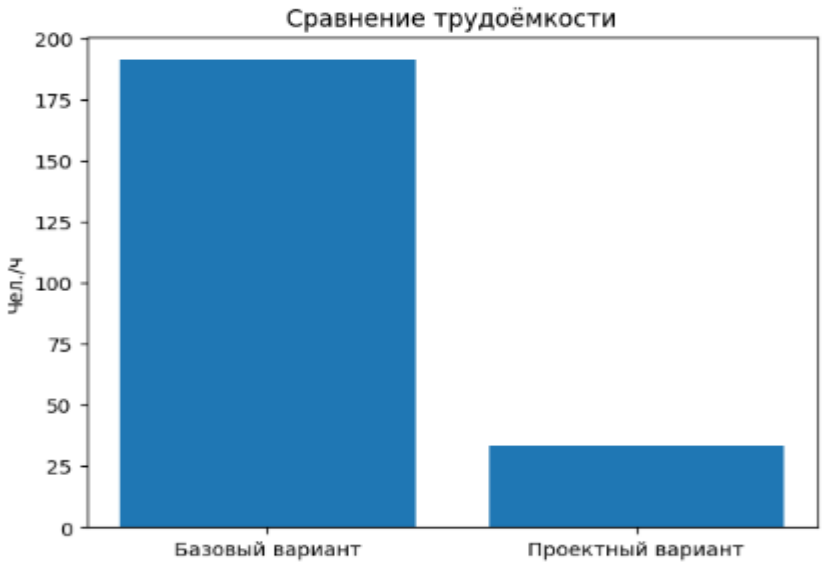


Рисунок 19 - Сравнение трудоёмкости выполнения операций при базовом и проектном вариантах

На рисунке 19 представлено сравнение трудоёмкости выполнения операций при базовом и проектном вариантах решения задачи. Внедрение информационной системы позволяет сократить трудоёмкость обработки информации более чем в 5 раз, что свидетельствует о высокой степени автоматизации бизнес-процессов.

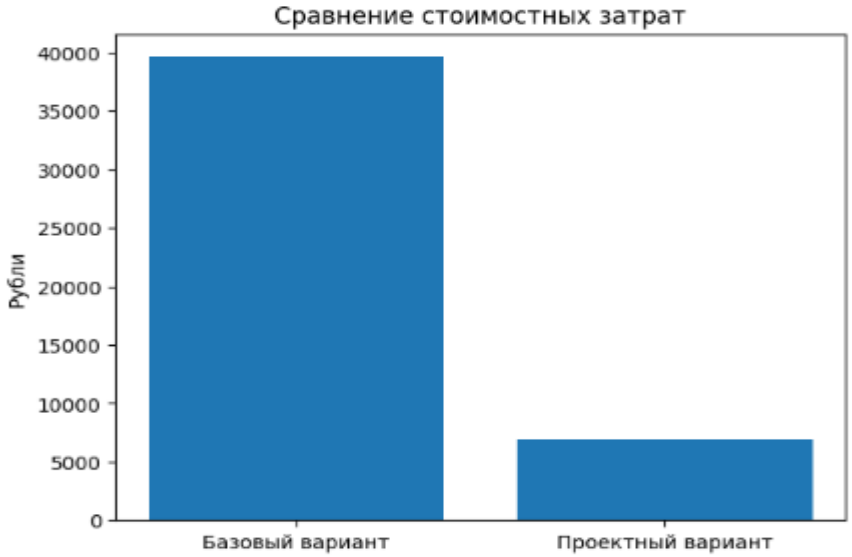


Рисунок 20 - Сравнение стоимостных затрат выполнения операций при базовом и проектном вариантах

Диаграмма, представленная на рисунке 3.2, отражает снижение стоимостных затрат на обработку информации при использовании проектного решения. Экономия составляет 32 707,48 руб. в год, что подтверждает экономическую целесообразность внедрения разрабатываемого модуля.

Для расчёта стоимостных затрат рассчитывается среднечасовая норма оплаты труда для сотрудника Компании. Исходя из средней 22-дневной рабочей недели (8-часовой рабочий день) и средней месячной зарплаты 36500 руб. получается:

$$N_З = 36500 \text{ руб.} / (22 \text{ д.} * 8 \text{ ч.}) = 207{,}39 \text{ руб./ч.}$$

В таблице 16 представлены расчеты трудовых и экономических затрат при базовом и проектном варианте решения задачи. Коэффициент изменения трудовых и стоимостных затрат показывает снижение затрат после внедрения информационной системы и составляет 0,17, сокращение затрат прогнозируется на 83 %. Индекс изменения затрат, равный 5,73, указывает на то, что затраты в базовом варианте превышают проектный вариант более чем в пять раз.

Полученные значения являются взаимно обратными и подтверждают значительный экономический эффект от внедрения проекта.

Таблица 16

Расчёт показателей эффективности от 7 едрения проекта

	Затраты		**Снижение затрат**	**Коэффициент изменения трудовых затрат**	**Индекс изменения трудовых затрат**
	Базовый вариант	**Проектный вариант**			
Трудоёмкость	T_0 (ч)	T_1 (ч)	$\Delta T = T_0 - T_1$ (ч)	$K_T = \frac{\Delta T}{T_0}$	$I_T = \frac{T_0}{T_1}$
	191,05	33,34	157,71	0,17	5,73
Стоимость	C_0 (руб.)	C_1 (руб.)	$\Delta C = C_0 - C_1$ (руб.)	$K_C = \frac{\Delta C}{C_0}$	$I_C = \frac{C_0}{C_1}$
	39621,86	6914,38	32707,48	0,17	5,73 1

Данные таблицы 10 говорят о значительном сокращении трудовых и стоимостных затрат. Сокращение трудовых затрат в абсолютном выражении составило 157,71 чел./ч, стоимостных затрат — 32707,48 руб. в год.

Затраты на создание проекта рассчитаны в п. 3.1 и составляют

$К_п$ = 85941,40 руб.

Срок окупаемости проекта рассчитывается:

$Т_{ок}$ = 85941,40 руб. / 32707,48 руб. ≈ 2,63 лет ≈ 31 мес.

Расчетный коэффициент эффективности капитальных затрат:

$Е_р = 1 / Т_{ок} = 1 / 2,63 = 0,38$

Годовой экономический эффект равен годовому снижению стоимостных затрат:

$Э = П - К_П \times E_н$, где:

$Э$ – годовой экономический эффект;

$П$ – годовая экономия (годовой прирост прибыли, Δ С), руб.;

$К_П$ – единовременные капитальные затраты (на программирование, проектирование, отладку и внедрение), руб.;

$Е_н$ – нормативный коэффициент эффективности капитальных вложений. (минимальная норма эффективности капитальных вложений, ниже которой они нецелесообразны). Значение $Е_н$ принимается равным 0,15.

Э = 32707,48 руб. – 85941,40 руб. * 0,15 = 19816,31 руб.

Помимо количественного экономического эффекта, также важно учитывать и качественные изменения в работе отдела по обработке входящих заявок. Такой эффект выражен появлением новых средств и способов обработки информации, которые предоставляют новые возможности:

- централизованного учета информации и получения доступа к любым выборкам данных;
- статистического анализа деятельности по обработке входящих заявок;
- автоматизированной проверки ввода информации;
- разграничения прав доступа к данным.

Выводы по главе 1 проведено экономическое обоснование эффективности внедрения разработанной информационной системы доставок управленческого учёта. В качестве методической основы использован сравнительный анализ базового и проектного вариантов обработки информации, основанный на расчёте трудоёмкости и стоимостных затрат. Выполненные расчёты показали снижение трудоёмкости выполнения операций после внедрения автоматизированной системы — более чем в пять раз по сравнению с базовым вариантом.

Снижение трудовых затрат привело к уменьшению годовых стоимостных затрат на обработку информации на 32707,48 руб., что подтверждает экономическую целесообразность проекта. Рассчитанный срок окупаемости проекта составил около 2,6 лет. Кроме количественного экономического эффекта, внедрение системы обеспечивает существенные качественные преимущества: повышение оперативности принятия управленческих решений, снижение риска ошибок, улучшение контроля дебиторской задолженности.

ЗАКЛЮЧЕНИЕ

Дипломная работа была посвящена решению актуальной задачи автоматизации системы бронирования билетов компании ООО «Самокат» в условиях цифровой трансформации бизнес-процессов.

В ходе выполнения работы достигнута поставленная цель — разработана и обоснована автоматизированная информационная система, обеспечивающая централизованное хранение, обработку и анализ онлайн-бронирования билетов. Последовательно решены все задачи, сформулированные во введении: проведён анализ деятельности предприятия и существующих бизнес-процессов, выполнен выбор стратегии автоматизации, спроектированы информационное и программное обеспечение системы, реализована база данных и программные модули, а также обоснована экономическая эффективность проекта.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Александров, О.А. Экономический анализ: Учебное пособие / О.А. Александров, Ю.Н. Егоров. -М.: ИНФРА-М, 2016. - 288 с.

2. Б.Я. Советов, А.И. Водяхо, В.А. Дубенецкий, В.В. Цехановский. Архитектура информационных систем. – М.: Академия, 2021. – 288 с.

3. Б.Я. Советов, В.В. Цехановский. Информационные технологии. – М.: Юрайт, 2021. – 272 с.

4. Балдин К.В., Уткин В.Б. Информационная система розничной торговли в экономике. — М.: ИнфоАрт, 2009. —395 с.

5. Бреусов, В.А. Эффективные закупки. Анализ ассортимента. - М.:Высшая школа корпоративного управления, 2018г. -161 с.

6. Бурова, Е. М. Справочник по архивному делу для работников государственных органов, органов местного самоуправления и организаций / Е. М. Бурова, Н. А. Муравьева. – Москва : Общество с ограниченной ответственностью (ООО) «ТЕРМИКА.РУ», 2025. – 458 с.

7. Г.Сенкевич. Информационная система малого предприятия "с нуля". Самое необходимое. – СПб.: БХВ-Петербург, 2021. – 400 с.

8. Галямина И.Г., «Управление процессами» [Текст], - СПб.: Питер, 2013. - 304 с.

9. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания: утв. постановлением N 472 от 29.12.1998 // Консультант плюс: справочно-правовая система.

10. Дейт К. Дж., «Введение в системы баз данных» [Текст], 8-е издание.: Пер. с англ. — М.: Издательский дом “Вильяме”, 2005. — 1328 с.: ил. — Парал. тит. англ.

11. Замыцкий, П. В. Автоматизация принятия управленческих решений при оперативном учете хода производства на основе систем электронного документооборота : специальность 05.13.06 "Автоматизация и управление технологическими процессами и производствами (по отраслям)" : диссертация

на соискание ученой степени кандидата технических наук / П. В. Замыцкий. – Москва, 2013. – 154 с.

12. Замыцких, П. В. Автоматизация принятия управленческих решений при оперативном учете хода производства на основе систем электронного документооборота : специальность 05.13.06 "Автоматизация и управление технологическими процессами и производствами (по отраслям)" : автореферат диссертации на соискание ученой степени кандидата технических наук / Замыцких Павел Викторович. – Москва, 2013. – 24 с.

13. Зараменских, Е.П. Управление жизненным циклом информационных систем - М.:Юрайт, 2018. – 431 с.

14. Информатика: Учебник / Под ред. проф. Н. В. Макаровой. — М.: Финансы и статистика, 2005. — 768 с.

15. Коцюба И.Ю. «Основы проектирования информационных систем» [Текст]/ И.Ю. Коцюба, Чунаев А.В., А.Н. Шиков – Санкт-Петербург: Изд-во ИТМО, 2015 г, 206 стр.

16. Кривокоченко Л. В. Научная электронная библиотека. Мировые товарные рынки. — https//cyberleninka.ru/article/n/mirovoy-rynok-mebeli-i-pozitsii-na-nem-rossii-v-svete-prisoedineniya-k-vto.

17. Культин Н. Б. «MS Project в задачах и примерах.» - СПб.: БХВ-Петербург, 2019. - 320 с.

18. Лубяная, Ю. А. Первичный учет и документооборот как составляющие правового обеспечения принятия управленческих решений / Ю. А. Лубяная, Н. П. Король // Научный вектор : Сборник научных трудов. – Ростов-на-Дону : Ростовский государственный экономический университет "РИНХ", 2024. – С. 123-125.

19. Маклаков С.В., Туманов В.Е., «Проектирование реляционных хранилищ данных» [Текст], - М.: Диалог-МИФИ, 2007, - 336 с.

20. Малашук, Н.М. Внешняя среда предприятия. / Н.М. Малашук - Текст: электронный // Экономические науки: интернет-портал. – URL:

https://cyberleninka.ru/article/n/vneshnyaya-sreda-predpriyatiya/viewer (дата обращения: 23.04.2022). – Текст: электронный.

21. Медведева, О. В. Терминология документоведения / О. В. Медведева // Актуальные вопросы науки и практики и перспективы их решений : Сборник научных трудов по материалам VIII Международной научно-практической конференции, Анапа, 05 июля 2025 года. – Анапа: ООО "Научно-исследовательский центр экономических и социальных процессов" в Южном Федеральном округе, 2025. – С. 99-103.

22. Мезенцев К.Н. Автоматизированные информационные системы. – М.: Академия, 2021. – 174 с.

23. Моделирование бизнеса. — https://analytics.infozone.pro/business-modeling-idef-umlaris.

24. Нагорная, М. Л. Состав и содержание архивных терминов в проекте нового ГОСТ Р система стандартов по информации, библиотечному и издательскому делу (СИБИД). Делопроизводство и архивное дело. Термины и определения / М. Л. Нагорная // Документ. Архив. Информационное общество : Материалы VI Научно-практической конференции с международным участием, Москва, 26–27 октября 2023 года. – Москва: ООО "ТЕРМИКА.РУ", 2024. – С. 330-341.

25. Научная электронная библиотека. Вопросы экономических наук. — https://cyberleninka.ru/article/n/vneshnyaya-sreda-predpriyatiya/viewer.

26. Оценки экспертов FAO. The UNECE/ FAO Forest Products Annual Market Review. — https//unece.org/trade/timber.

27. Оценки экспертов Ассоциации мебельной промышленности Германии. — https//: hdh-ev.de/german .

28. Партыка Т.Л., Попов И.И. Информационная безопасность. – М.: Форум, Инфра-М, 2002. – 368 с.

29. Плотников В. CRM: с чего начать (взгляд разработчика). — http: //www.crmonline.ru/analytics/experts/plotn2.php.

30. Поддевалина, Я. А. трактовка понятия «электронный документооборот» / Я. А. Поддевалина // Пичетовские чтения - 2024. История: наука, образование, память : Материалы международной научно-практической конференции, посвященной 90-летию исторического факультета Белорусского государственного университета, Минск, 25 октября 2024 года. – Минск: Белорусский государственный университет, 2024. – С. 459-463.

31. С.Г. Пудовкина «Анализ и оптимизация бизнес-процессов» [Текст], Челябинск, 2013, 51 с.

32. Семакин И.Г. Хеннер Е.К. Информационные системы и модели. — М.: Бином, 2005. 303 с.

33. Стерхов А. П. Комплексная безопасность бизнеса и основные проблемы её обеспечения. — https://cyberleninka.ru/article/n/kompleksnaya-bezopasnostbi znes a-i-osnovnye-problemy-eyo-obespecheniya.

34. Ховрич, А. Д. Организация документооборота в условиях взаимосвязи бухгалтерского и управленческого учёта / А. Д. Ховрич // ЭКОНОМИКА в ТЕОРИИ и на ПРАКТИКЕ: АКТУАЛЬНЫЕ ВОПРОСЫ и современные АСПЕКТЫ : сборник статей XIV Международной научно-практической конференции, Пенза, 25 октября 2022 года. – Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2022. – С. 51-54.

35. Шнайдер Д.А., Подход к оперативному управлению предприятием/ Д.А. Шнайдер// ВестникЮУрГУ. - 2019. № 13. - С. 70–73.

Дипломная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной литературы и других источников имеют ссылки на них. Выпускная квалификационная работа прошла проверку на корректность заимствования в системе «Антиплагиат.ВУЗ».

Настоящим подтверждаю, что даю разрешение Университету «Синергия» на размещение полного текста моей выпускной квалификационной работы и отзыва о работе в период ее подготовки в электронно-библиотечной системе Университета «Синергия».

_____ Лифанов Р.В.
подпись Фамилия ИО

«___» _____ 20__ г.