



**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технологический университет «СТАНКИН»**  
**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

**Институт  
Информационных  
технологий**

---

**Кафедра  
Информационных технологий  
и вычислительных систем**

**ПРОЕКТ ПО ДИСЦИПЛИНЕ «Технологии индустрии 4.0»**

**СТУДЕНТОВ 3 КУРСА бакалавриата ГРУППЫ ИДБ-22-05**  
(уровень профессионального образования)

**Павлова Артема  
Недовича Павла**

---

**НА ТЕМУ**

**« Web-приложение по подбору мест отдыха и досуга»**

---

**Направление: 09.03.01 Информатика и вычислительная техника**

**Преподаватель: Волкова О. Р., Доцент**  
(Ф.И.О., должность, степень, звание)

\_\_\_\_\_  
(подпись)

**МОСКВА 2025**

## Оглавление

2. Выбор технологического стека и обоснование .....	3
2.1. Фреймворк: Next.js .....	3
2.2. Библиотека: React .....	3
2.3. Язык программирования: TypeScript .....	3
2.4. UI-компоненты: Shadcn .....	4
2.5. База данных: PostgreSQL с провайдером Neon .....	4
2.6. Интеграция искусственного интеллекта: Gemini .....	4
2.7. Инструменты для разработки и управления проектом .....	5
2.8. Вывод .....	5

## 2. Выбор технологического стека и обоснование

Для успешной реализации проекта по созданию интерактивного веб-приложения для поиска и управления местами (рестораны, кафе и другие заведения) использованы современные веб-технологии, обеспечивающие высокую производительность, удобный интерфейс и надежное хранение данных. В данной главе рассматриваются основные компоненты технологического стека и обоснование их выбора.

### 2.1. Фреймворк: Next.js

Для разработки веб-приложения выбран фреймворк Next.js, основанный на React. Данный выбор обусловлен следующими факторами:

Серверный рендеринг и статическая генерация: Next.js поддерживает серверный рендеринг (SSR) и статическую генерацию страниц (SSG), что улучшает производительность и SEO.

Гибкость и масштабируемость: Возможность создания как статических, так и динамических страниц позволяет адаптировать приложение под различные сценарии использования.

Встроенная маршрутизация: Next.js предоставляет файловую систему маршрутов, упрощающую организацию страниц и API-роутов.

Экосистема и сообщество: Большое сообщество и богатая документация ускоряют разработку и решение проблем.

Поддержка TypeScript: Встроенная поддержка TypeScript обеспечивает строгую типизацию и снижает вероятность ошибок.

### 2.2. Библиотека: React

Для построения интерактивного пользовательского интерфейса использована библиотека React. Обоснование выбора:

Компонентный подход: React позволяет создавать переиспользуемые UI-компоненты, что упрощает разработку и поддержку кода.

Эффективное управление состоянием: Хуки и контекст React обеспечивают удобное управление состоянием приложения.

Широкая экосистема: Большое количество библиотек, таких как React Hook Form и Zod, упрощают реализацию форм и валидации.

Высокая производительность: Виртуальный DOM минимизирует количество обновлений реального DOM, обеспечивая быструю отрисовку интерфейса.

### 2.3. Язык программирования: TypeScript

Для написания кода использован TypeScript, добавляющий строгую типизацию к JavaScript. Основные преимущества:

Снижение ошибок: Строгая типизация позволяет выявлять ошибки на этапе компиляции, улучшая качество кода.

Улучшенная поддержка инструментов: TypeScript обеспечивает автодополнение и проверку типов в редакторах кода, ускоряя разработку.

Совместимость с JavaScript: TypeScript полностью совместим с существующим JavaScript-кодом и библиотеками, такими как React и Next.js.

Поддержка сложных структур данных: Удобство работы с типами данных, таких как объекты мест, пользовательские профили и отзывы.

## 2.4. UI-компоненты: Shadcn

Для создания стильного и доступного интерфейса использована библиотека компонентов Shadcn. Обоснование выбора:

Готовые компоненты: Shadcn предоставляет набор современных, настраиваемых UI-компонентов, таких как карточки, кнопки и формы.

Интеграция с Tailwind CSS: Компоненты оптимизированы для работы с Tailwind CSS, что упрощает стилизацию и обеспечивает единообразный дизайн.

Доступность: Компоненты соответствуют стандартам доступности (a11y), улучшая пользовательский опыт для всех категорий пользователей.

Настраиваемость: Возможность адаптации компонентов под конкретные требования дизайна проекта.

## 2.5. База данных: PostgreSQL с провайдером Neon

Для хранения данных пользователей, истории поиска, избранных мест и отзывов использована реляционная база данных PostgreSQL с облачным провайдером Neon. Основные преимущества:

Надежность и масштабируемость PostgreSQL: Высокая производительность и поддержка больших объемов данных.

Облачные возможности Neon: Автоматическое масштабирование, серверлесс-архитектура и упрощенное управление базами данных, что снижает затраты на обслуживание.

Поддержка сложных запросов: Возможность выполнения сложных SQL-запросов для фильтрации и агрегации данных о местах и отзывах.

Безопасность: Встроенные механизмы управления доступом и шифрования защищают пользовательские данные.

Совместимость с Prisma: Использование ORM Prisma упрощает взаимодействие с базой данных и обеспечивает типобезопасность.

## 2.6. Интеграция искусственного интеллекта: Gemini

Для обработки пользовательских запросов и улучшения функциональности приложения использована модель искусственного интеллекта Gemini. Обоснование выбора:

Высокая производительность: Gemini обеспечивает быструю и точную обработку текстовых запросов, что улучшает поиск мест и персонализацию рекомендаций.

Гибкость: Поддержка различных сценариев использования, таких как анализ отзывов и генерация описаний мест.

Интеграция с веб-приложением: Простое подключение через API позволяет встроить возможности ИИ в интерфейс на базе Next.js и React.

Адаптивность: Возможность дообучения модели под специфические задачи проекта, такие как классификация пользовательских предпочтений.

## 2.7. Инструменты для разработки и управления проектом

Для эффективной разработки, контроля версий и управления проектом использованы:

Git и GitHub: Для хранения кода, совместной работы и управления версиями.

Vercel: Для деплоя и хостинга приложения, с поддержкой автоматического масштабирования и CI/CD.

## 2.8. Вывод

Выбранный технологический стек (TypeScript, Next.js, React, Shadcn, PostgreSQL с Neon, Gemini и сопутствующие инструменты) позволяет создать производительное, масштабируемое и интерактивное веб-приложение для поиска и управления местами. Next.js и React обеспечивают гибкость и высокую скорость работы интерфейса, TypeScript повышает надежность кода, Shadcn ускоряет разработку стильного UI, PostgreSQL с Neon гарантирует надежное и масштабируемое хранение данных, а Gemini добавляет интеллектуальные возможности. Данный подход позволяет разработать эффективное приложение, соответствующее современным требованиям.