

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ  
Школа бакалавриата

## ОТЧЕТ

По проекту  
«Разработка веб-сайта для организационного психолога с системой  
администрирования»

по дисциплине «Проектный практикум»

Заказчик: Денисюк В.В.

Куратор: Денисюк В.В.

ученая степень, ученое звание, должность

Студенты команды Miike

Мокрушина А.С.

Ижбулатова Д.Э.

Емельянова М.В.

Кудря А.С

ООО "66 Бит"

---

---

---

---

---

---

---

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Целевая аудитория .....	4
2 Анализ аналогов .....	5
3 Функциональные требования .....	9
3.1 CMS .....	9
3.2 Функционал пользователя .....	9
3.3 Интеграция с Telegram .....	9
4 Технические требования .....	10
4.1 Архитектура и технологии .....	10
4.2 Производительность .....	10
4.3 Безопасность .....	10
4.4 SEO .....	10
4.5 Адаптивность и кроссбраузерность .....	10
4.6 Хранение данных .....	11
5 Требования к дизайну .....	12
5.1 Общий стиль .....	12
5.2 Адаптивность .....	12
5.3 Контент .....	12
6 Структура сайта .....	13
7 Архитектура программного обеспечения .....	18
8 Методология разработки .....	21
9 Отчеты каждого из участников .....	24
9.1 Тимлид и аналитик (Мокрушина А.С.) .....	24
9.2 Дизайнер (Ижбулатова Д.Э.) .....	25
9.3 Бэкенд-разработчик (Емельянова М.В.) .....	26
9.4 Фронтенд-разработчик (Кудря А.С.) .....	28
ЗАКЛЮЧЕНИЕ .....	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	31
ПРИЛОЖЕНИЕ А Артефакты проекта .....	32

## ВВЕДЕНИЕ

Тип сайта: персональный сайт психолога, многостраничный

Цель и задачи сайта: привлечение клиентов, предоставление услуг специалиста, упрощение записи на консультации, минимизация нецелевых или нерелевантных сообщений, предоставление полезной информации, сбор заявок и вопросов от заинтересованных лиц.

Сайт предназначен для личного пользования сотрудника компании.

Веб-сайт предназначен для продажи услуг организационного психолога.

Актуальность создания персонального сайта психолога обусловлена необходимостью эффективного продвижения и реализации профессиональных услуг в современной цифровой среде. Заказчица подчеркнула, что ключевым аспектом для нее является доступность и ясность информации для клиентов. Пользователи должны легко понимать, какие услуги предлагает специалист, как они предоставляются, и каким образом осуществляется запись на консультации.

Требуемый результат: Веб-сайт с системой администрирования для управления контентом.

Уровни результата (минимальный / базовый / оптимальный):

- 1) Минимальный: Сайт с базовым функционалом (список услуг, форма записи, система администрирования).
- 2) Базовый: Добавлена валидация формы, интеграция с Telegram, SEO-оптимизация.
- 3) Оптимальный: Запущенный сайт с защитой от спама, современным дизайном и полным набором функций.

## 1. Целевая аудитория

### Физические лица:

- заинтересованные в организационной психологии,
- нуждающиеся в карьерном консультировании,
- занимающие руководящие должности,
- имеющие выгорание и стресс, связанный с работой.

### Юридические лица:

- желающие провести диагностику команд и процессов компании,
- нуждающиеся в консультациях по подбору, мотивации сотрудников.

## 2. Анализ аналогов

### 1) [Mypsy.org](https://myspy.org)

Сайт имеет чистый и профессиональный дизайн, использующий спокойную цветовую палитру, что помогает создать доверие и авторитет — это важно для психологической практики.

Структура сайта ясная, разделы для услуг, отзывов и контактных форм хорошо видны, что способствует вовлечению клиентов.

Основные функции: Легкая навигация, быстрый доступ к услугам, биографии специалиста и блогу. Кнопки призыва к действию (СТА) расположены удачно, что улучшает конверсию.

Преимущества:

- ясная и интуитивно понятная навигация,
- профессиональный дизайн, который способствует доверию,
- разделы услуг и статьи, которые обучают и приносят ценность.

Недостатки:

- контент выглядит немного универсально,
- нет персонального подхода к клиентам,
- для подачи заявки или вопроса нужно переходить на сторонние ресурсы.

### 2) [Marinaionycheva.ru](https://marinaionychева.ru)

Используется яркая цветовая схема и динамичные элементы, что делает сайт более дружелюбным и вовлекающим.

Сайт активно использует клиентские отзывы и личный брендинг, что эффективно для создания доверия среди клиентов.

Основные функции: Четкие разделы для услуг и контактной информации, прямой подход к записи на консультации.

Преимущества:

- акцент на персонализацию и подробное представление специалиста,
- отзывы клиентов помогают укрепить доверие,
- удобные формы для связи и записи.

Недостатки:

- дизайн может не подойти всем, особенно тем, кто ищет более официальный и корпоративный стиль,
- нужно больше образовательного контента или статей о предлагаемых услугах,
- мало информации о психологах.

### 3) [Zigmund.online](https://Zigmund.online)

Сайт имеет современный дизайн с четкой структурой, где есть разделы для услуг, биографии специалиста и образовательных материалов.

Основные функции: Простой доступ к услугам и записи на консультации. Прямой подход с акцентом на то, что пользователи должны знать о терапии.

Преимущества:

- четкая структура с легко доступными услугами и возможностью записи,
- современный, привлекательный дизайн,
- образовательный контент подается в удобном формате,
- наличие онлайн-тестов.

Недостатки:

- чтобы задать вопрос, нужно переходить на сторонний ресурс.

### 4) [Lifshite.ru](https://Lifshite.ru)

Сайт создан на тильде и использует лаконичный дизайн.

Основные функции: Прямые сообщения и предложение услуг, акцент на личностное развитие и стресс-менеджмент.

Преимущества:

- просто и легко подана информация.

Недостатки:

- мало образовательного контента, который мог бы помочь пользователям узнать больше о психологических услугах,
- чтобы задать вопрос нужно переходить на сторонний ресурс.

#### 5) [Detok.org](https://detok.org)

Сайт более традиционный, спокойный, с акцентом на образовательные материалы и услуги.

Структура сайта ясная, важные разделы для записи и контактов легко доступны.

Основные функции: Сайт ориентирован на детоксикацию и развитие личности, предоставляя информацию о предлагаемых услугах.

Преимущества:

- профессиональный и спокойный дизайн, ориентированный на информацию и услуги,
- легко ориентироваться, есть формы для записи и контактная информация.

Недостатки:

- дизайн достаточно базовый и может нуждаться в более современных элементах,
- отсутствие интерактивных функций, таких как анкеты или инструменты для вовлечения посетителей,
- сложно разобраться с функционалом сайта,
- чтобы задать вопрос нужно звонить.

На основе анализа конкурентов можно выделить ключевые преимущества сайта:

- 1) Профессиональный и понятный дизайн с четкой структурой и удобными формами записи.
- 2) Удобство взаимодействия без необходимости переходить на сторонние ресурсы.

- 3) Индивидуальный подход и ориентация на работу одного специалиста.
- 4) Интеграция с Telegram для оперативной обратной связи.

Таким образом, сайт сочетает сильные стороны конкурентов и решает их недостатки, предлагая клиентам комфорт и оперативность.



### 3. Функциональные требования

#### 3.1 CMS (Система администрирования)

- добавление / редактирование / удаление: информации о себе, услуг, статей, контактов,
- просмотр заявок и вопросов,
- изменение статуса заявок и вопросов,
- возможность экспорта заявок и вопросов экспорта в Excel.

#### 3.2 Функционал пользователя

- просмотр списка услуг, статей, информации о специалисте,
- возможность перехода между страницами,
- заполнение и подача форм заявки и вопроса,
- без регистрации.

#### 3.3 Интеграция с Telegram

- обе формы (заявка/вопрос) отправляют данные в Telegram-бот администратора или в нужный чат.

## 4. Технические требования

### 4.1 Архитектура и технологии

- сайт должен быть реализован на Django, кастомный React с бэкендом,
- поддержка CMS с возможностью редактирования контента (услуги, статьи, информация о специалисте, контакты),
- база данных: SQLite / PostgreSQL / MySQL,
- панель администратора с защищённым доступом посредством ограничения паролем.

### 4.2 Производительность

- загрузка страниц < 2 секунд (на 4G).

### 4.3 Безопасность

- обязательное подключение HTTPS (SSL),
- валидация форм на клиенте и сервере,
- защита от спама (опционально): honeypot-поле, простая CAPTCHA,
- админка не индексируется (robots.txt, noindex),
- защита от XSS, CSRF, SQL-инъекций (опционально).

### 4.4 SEO

- ЧПУ (человекопонятные урлы) — на всех страницах, услугах и статьях,
- возможность задавать вручную: <title>, <meta name="description">, <meta property="og:...">.

### 4.5 Адаптивность и кроссбраузерность

- полная адаптивная верстка (мобильные, планшеты, десктопы),

- поддержка всех современных браузеров: Chrome, Firefox, Safari, Edge, Android/iOS.

#### 4.6 Хранение данных

- все заявки и вопросы сохраняются в бд или excel/csv),
- просмотр в админке через карточки заявок и вопросов,
- экспорт — вручную через интерфейс администратора.

## 5. Требования к дизайну

### 5.1 Общий стиль

- фиолетовый — акцентный цвет,
- минималистичный, не перегруженный визуальными элементами,
- шрифты читаемые, без засечек

Разный размер и вес шрифта для визуальной иерархии:

Крупные заголовки (например: 32–48px)

Тексты (14–18px)

Подписи / служебный текст (12–14px),

- просторные отступы между блоками (не «всё в кучу»),
- иконки — плоские, простые, без лишнего объёма.

### 5.2 Адаптивность

- полная адаптация под: смартфоны, планшеты и десктопы,
- кнопки и поля формы должны быть удобны для нажатия пальцем.

### 5.3 Контент

- на этапе разработки используется заглушечный текст,
- все тексты предоставляются заказчиком.

## 6. Структура сайта

### 1) Шапка сайта (header)

Содержит:

- логотип (при наличии) находится в левом верхнем углу,
- элементы навигации (кнопки): главная, о специалисте, статьи, услуги, заявка, задать вопрос.

### 2) Главная страница (main)

- блок 1: информация о Василине Денисюк

Имеет навигационную кнопку «узнать больше», которая открывает страницу «О специалисте»,

- блок 2: информация об организационной психологии

Содержит краткие выжимки из статей о данной психологии

Имеет навигационную кнопку, которая открывает страницу «Статьи»,

- блок 3: виды услуг

Имеет переключатель для разделения физических и юридических лиц

Содержит неполную информацию о разновидностях предоставляемых услуг

Имеет навигационную кнопку «узнать подробнее», которая открывает страницу «Услуги»,

- блок 4: содержит две кнопки навигации, первая направляет на форму создания заявки, вторая – на форму создания вопроса.

### 3) Страница «О специалисте»

Содержит информацию:

- фото,
- опыт работы,
- образование,

- методы работы,
- цитату.

#### 4) Страница «Статьи»

Имеет два раздела: книжную полку, где специалист рассказывает своё мнение о прочитанных книгах и делится своими мыслями на их счёт статьи, и статьи на разные темы

- теги у статей,
- лента статей (заголовок, краткое описание, дата),
- по умолчанию видно первые 5 статей,
- полная статья открывается на отдельной странице,
- в конце кнопка «показать больше»

Книжная полка содержит:

- управление финансами,
- тайм менеджмент,
- самодиагностика и управление состоянием.

#### 5) Страница с услугами

Разделение на 2 блока по типу клиента (Физическое/юридическое лицо)

Каждая услуга имеет:

- название,
- продолжительность,
- подробное описание,
- стоимость,
- кнопку «Записаться» (форма с заявкой).

#### 6) Страница с вопросами

Имеет: интересные вопросы и ответы на них

## 7) Форма с заявкой

### Обязательные:

- тип клиента (Физическое/Юридическое лицо),
- ФИО (оба типа),
- номер телефона (оба типа),
- email (оба типа),
- должность (юр. лица),
- название компании (юр. лица),
- ИНН (юр. лица),
- предпочитаемый способ связи (оба типа),
- описание запроса (оба типа),
- согласие на обработку персональных данных (оба типа).

### Опциональные:

- выбранная услуга (оба типа),
- должность (физ. лица),
- желаемые дата и время консультации (оба типа),
- дополнительные файлы (оба типа).

## 8) Форма с вопросом

### Обязательные:

- тип клиента (Физическое/Юридическое лицо),
- ФИО (оба типа),
- номер телефона (оба типа),
- email (оба типа),
- тема вопроса (оба типа),
- должность (юр. лица),
- ИНН (юр. лица),
- предпочитаемый способ связи (оба типа),
- описание ситуации (оба типа),

- согласие на обработку персональных данных (оба типа).

Опциональные:

- должность (физ. лица),
- дополнительные файлы (оба типа).

## 9) Архив заявок

Доступен только через административный интерфейс

Представление в виде карточек

Каждая карточка содержит:

- ФИО,
- компания (если указана),
- телефон,
- дата и время отправки,
- статус (например: Новая, В работе, Обработана, Отклонена),
- Краткий фрагмент описания запроса.

Фильтрация и сортировка:

- по дате,
- по статусу,
- по типу клиента (физлицо / юрлицо).

При открытии карточки видна вся информация (всплывающее окно или отдельная страница)

## 10) Архив вопросов (аналогично архиву с заявками)

Доступен только через административный интерфейс

Представление в виде карточек

Каждая карточка содержит:

- номер телефона,
- дата отправки,
- фрагмент вопроса



- статус (например: Новая, В работе, Обработана, Отклонена).

Фильтрация и сортировка:

- по дате,

- по статусу,

- по типу клиента (физлицо / юрлицо).

При открытии карточки видна вся информация (всплывающее окно или отдельная страница)

## 11) Подвал сайта (footer)

Содержит:

- контакты,

- список разделов.

## 7. Архитектура программного обеспечения

Бэкенд (серверная часть): Реализован на Django и Django REST Framework, обеспечивает бизнес-логику, хранение данных и API для взаимодействия с фронтендом. Основные приложения включают:

- main: управляет информацией о специалисте,
- articles: отвечает за управление статьями, их содержимым, датами публикации и тегами,
- services: определяет услуги, предоставляемые специалистом, с их названием, описанием, стоимостью и типом клиента,
- requests: обрабатывает заявки от клиентов на консультацию/услугу, включая контактные данные, тип запроса, статус и информацию о выбранной услуге. Имеет внешний ключ на компонент services, что позволяет связать заявку с конкретной услугой,
- questions: Управляет вопросами, поступающими от клиентов, содержащими контактные данные, текст вопроса, тему, статус и ответ специалиста.

Административный интерфейс (Django Admin): Предоставляет удобный инструмент для управления содержимым сайта и обработки запросов/вопросов. Доступен только авторизованным администраторам. Функционал включает:

- добавление, редактирование, удаление услуг, статей, информации о специалисте,
- просмотр архивов заявок и вопросов,
- изменение статуса заявок и вопросов,
- фильтрация и сортировка заявок и вопросов по различным критериям (дата, статус, тип клиента),
- возможность экспорта данных заявок и вопросов в формат Excel.

Интеграция с Telegram: реализована отправка уведомлений о новых заявках и вопросах в Telegram-бот администратора, что обеспечивает оперативное реагирование.

Выбор архитектуры основан на необходимости:

- быстро развертываемой и масштабируемой среды,
- модульного подхода для независимой разработки и тестирования компонентов,
- наличия встроенной системы администрирования (Django Admin),
- безопасной обработки персональных данных,
- простоты в реализации REST API для интеграции с внешними системами (например, Telegram).

Использование Django обеспечивает устойчивую архитектурную основу и поддержку всех требуемых компонентов «из коробки», включая ORM, маршрутизацию, шаблоны, администрирование и сериализацию.

Фронтенд разработан на языке TypeScript (.ts, .tsx), использован фреймворк Next.js 15.3.0, библиотеки: React 19, React Router DOM, Axios.

Структура:

*app/* — маршруты и страницы:

- *layout.tsx* — основной шаблон (header, footer, стили, провайдеры),
- *page.tsx* — главная страница,
- *about/page.tsx* — страница «Обо мне»,
- *articles/page.tsx* — статьи,
- *questions/page.tsx* — часто задаваемые вопросы,
- *services/page.tsx* — услуги.

*components/* — переиспользуемые UI-блоки:

- Hero, Footer, Header — оформление главной страницы,

- AboutPage, ArticlesPage, QuestionsPage, ServicesPage — отдельные секции под каждую страницу,
- RequestForm, QuestionsForm — модальные формы,
- ServicesBusinesses, ServicesIndividuals, ServicesAndTeam — секции услуг, разделённые по типу клиента,
- OrgPsychology, TeamBoost — специальные секции для определённых направлений.

Каждый компонент имеет .tsx файл (логика и JSX) и .module.css файл (стилизация через CSS Modules).

Связи между компонентами:

- все страницы page.tsx импортируют соответствующие компоненты из components/,
- RequestForm используется на разных страницах как модальное окно (портал),
- layout.tsx задаёт общую структуру — подключает globals.css и общий UI
- компоненты подключаются по alias @/components/, API через @/api/.

Обоснование архитектурного выбора:

- Next.js выбран за SSR/SSG возможности и встроенный роутинг,
- компонентный подход — каждую секцию (About, Articles, Services) можно переиспользовать, независимо тестировать и менять,
- CSS Modules — изоляция стилей для каждого компонента,
- Axios — надёжная работа с REST API,
- TypeScript — строгая типизация и масштабируемость.

## 8. Методология разработки

Проект разработан с применением гибкой методологии разработки (Agile). Такой подход позволил эффективно управлять изменениями требований, обеспечивать регулярную обратную связь и оперативно реагировать на выявленные проблемы.

### 1) Процесс разработки бэкенда включал следующие этапы:

- Проектирование архитектуры и данных: на основе требований была разработана архитектура, определены основные приложения Django и их содержимое, связи между ними,

- последовательная реализация моделей и представлений,

- настройка административной панели,

- интеграция с Telegram.

### 2) Тестирование на промежуточных этапах

После завершения каждой значимой функциональной единицы или итерации проводилось тестирование. Это включало:

- интеграционное тестирование: проверка взаимодействия между различными модулями и приложениями (например, корректность сохранения данных формы в базе, отправка уведомлений в Telegram),

- функциональное тестирование: проверка соответствия реализованного функционала заданным требованиям, тестирование пользовательских сценариев.

### 3) Разбор выявленных ошибок

В ходе промежуточного тестирования было выявлено несколько типов ошибок, которые были успешно устранены.

Ошибки отображения данных в административном интерфейсе:

Изначально карточки заявок и вопросов в админке отображали только минимальный набор полей. Полная информация требовала перехода на отдельную страницу. Также отсутствовали удобные механизмы поиска и сортировки по всем необходимым полям.

Ошибки обработки файлов:

Прикрепленные файлы к заявкам и вопросам не сохранялись или были недоступны. Проблема была связана с некорректными настройками MEDIA\_ROOT и MEDIA\_URL в settings.py Django, а также с отсутствием необходимой конфигурации для отдачи статических и медиа-файлов в режиме разработки.

### 1) Подход и процесс фронтенд-разработки

Разработка ведётся по модульному принципу: каждый компонент инкапсулирует свою логику и стили. Все компоненты и страницы строго типизированы (.tsx, @types), использован strict режим TS. Используется портал для форм — показатель продвинутой работы с DOM (в частности, QuestionsFormPortal.tsx). Промежуточная проверка и отладка производится через next dev, а статика и сборка – через next build. ESLint работает на основе рекомендаций next/core-web-vitals.

### 2) Тестирование

- производилось вручную в dev-среде,
- TypeScript обеспечивает проверку на уровне компиляции,
- ошибки выявлялись в браузере и устранялись итеративно.

### 3) Предотвращение ошибок:

Конфликты CSS-классов между компонентами. Возникали при совпадении имён классов в разных стилевых модулях.

- Решение: применение технологии CSS Modules, обеспечивающей уникальность классов на уровне сборки.

Ошибки типизации в TypeScript. Обнаруживались при передаче пропсов без точного описания типов.

- Решение: введение интерфейсов и явное указание типов данных для всех компонентов и их параметров.

Неправильная отрисовка модальных окон. Проблема возникала при отображении форм вне основного DOM-дерева

– Решение: использование Portal API для безопасного рендеринга модальных компонентов.

Ошибки маршрутизации в App Router. Возникали при отсутствии файла page.tsx или неправильной структуре директорий

– Решение: пересборка проекта, проверка маршрутов и приведение структуры к требованиям Next.js.

Нарушение адаптивности интерфейса. Обнаруживалось при тестировании на устройствах с шириной экрана менее 1024px.

– Решение: внедрение медиазапросов и ручная проверка корректности отображения на всех целевых разрешениях.

## 9. Отчеты каждого из участников

### 9.1 Тимлид и аналитик (Мокрушина А.С.)

Работа тимлида началась с налаживании коммуникации с заказчицей и формирования четкого понимания того, каким она видит будущий продукт. Были организованы встречи, на которых подробно обсуждались требования, ожидаемые результаты и план разработки.

На протяжении всего процесса Анастасия отвечала за координацию работы команды, следила за дедлайнами и регулярно напоминала участникам о сроках, декомпозировала задачи для каждого члена команды. Для более эффективного управления задачами использовалась канбан-доска [Kaiten](#), преимуществом которой является уведомления в Telegram – каждый член команды получал сообщение при создании новой задачки или ее изменении.

В роли аналитика Анастасия начала с проведения анализ аналогов на рынке и определения целевой аудитории. По итогам анализа были выявлены сильные и слабые стороны существующих решений, что позволило определить приоритетные направления для разработки собственного продукта. Были разработаны схема структуры сайта, диаграммы взаимодействия пользователя и администратора с ресурсом, а также составлен файл с задачами и выполняемым функционалом. Благодаря этому удалось сформировать полное представление о сайте.

Следующим важным этапом стало составление технического задания, необходимого разработчикам для четкого понимания требований. После получения обратной связи от заказчицы техническое задание было полностью доработано с учетом дополнений.

В процессе работы аналитик использовала [Miro](#) (для создания схем и диаграмм), [Google Disk](#) (для составления письменных материалов проекта).



## 9.2 Дизайнер (Ижбулатова Д.Э.)

В рамках работы над проектом была выполнена последовательность этапов, направленных на разработку визуальной части сайта специалиста в области организационной психологии.

На подготовительном этапе проводился поиск визуальных и функциональных референсов, а также анализ сайтов схожей тематики. Целью данного этапа являлось формирование представления о профессиональных стандартах оформления, пользовательском опыте и актуальных решениях в сфере цифрового представления психологических услуг.

Был создан предварительный макет главной страницы, а также подготовлено несколько цветовых схем, представленных для согласования с заказчиком. После утверждения концепции макет главной страницы был доработан до полноценного дизайна с использованием временных текстовых и графических заглушек.

На следующем этапе велась разработка дизайнов дополнительных страниц — «О специалисте» и «Статьи». По завершении первого раунда проектирования был получен отзыв от заказчика, на основании которого в макеты были внесены необходимые изменения.

После утверждения структуры и визуальных решений заказчик предоставил текстовое наполнение для сайта. Соответственно, была проведена адаптация дизайна с учётом новых текстов, включая уточнение компоновки и стилистики блоков.

В дальнейшем были разработаны две отдельные формы: форма отправки вопросов и форма записи на консультацию. Обе формы были интегрированы в общий визуальный стиль сайта.

Заключительным этапом работы на текущем этапе стала разработка дизайна страницы со списком услуг, соответствующей общей логике структуры и стилистике сайта.

### 9.3 Бэкенд-разработчик (Емельянова М.В.)

Работа бэкенд-разработчика началась с анализа предоставленного технического задания, включающего описание функциональных требований, архитектуры и моделей данных. Проведено детальное изучение сущностей и их взаимосвязей, что позволило сформировать комплексное понимание структуры будущей базы данных и логики взаимодействия между компонентами.

В процессе подготовки был выбран следующий стек технологий: Django и Django REST Framework. Они были выбраны за их мощьность, гибкость и наличие готовых решений для быстрого прототипирования и разработки API. Проведена настройка базового Django-проекта, включая конфигурацию статических файлов, медиа-файлов и первоначальные настройки безопасности.

Ключевым этапом стала реализация моделей данных в соответствующих приложениях Django:

- в приложении main была создана модель `SpecialistInfo` для хранения всей информации о специалисте, включая фото, опыт, образование, социальные ссылки и реквизиты,
- приложение `articles` получило модель `Article` для управления статьями, с полями для заголовка, краткого и полного описания, даты публикации, статуса и тегов,
- для услуг было разработано приложение `services` с моделью `Service`, которая включает название, описание, стоимость, статус активности и тип клиента (физическое/юридическое лицо),
- приложение `requests` реализовало модель `ClientRequest` для обработки заявок, с обширным набором полей для контактных данных, типа клиента, описания запроса, статуса, предпочитаемого способа связи, а также специфичных данных для физических и юридических лиц, включая возможность прикрепления файлов,

– для вопросов было создано приложение `questions` с моделью `ClientQuestion`, аналогичной по структуре заявкам, но адаптированной под специфику вопросов и ответов, включая поля для темы вопроса и ответа специалиста.

Параллельно с разработкой моделей данных, были созданы соответствующие сериализаторы с использованием Django REST Framework для каждой модели. Сериализаторы обеспечивают корректное преобразование данных между форматом Python-объектов и JSON, а также отвечают за валидацию входящих данных. Была реализована кастомная валидация для обязательного согласия на обработку персональных данных в формах заявок и вопросов.

Следующим шагом стало создание API-представлений (`views`), использующих разработанные сериализаторы. Для каждой сущности были реализованы следующие типы представлений:

- `CreateAPIView` для создания новых записей (заявки, вопросы),
- `ListAPIView` для получения списков (статьи, услуги, заявки, вопросы),
- `RetrieveUpdateDestroyAPIView` для просмотра, обновления и удаления отдельных записей.

Одной из важных задач была интеграция с Telegram. Был использован вспомогательный модуль `utils.telegram_utils` для отправки уведомлений. В логику создания новых заявок и вопросов была добавлена функция отправки подробного сообщения в Telegram-бот администратора, содержащего всю необходимую информацию о запросе.

Для удобства управления данными была настроена административная панель Django. В файлах `admin.py` каждого приложения были зарегистрированы соответствующие модели. Для упрощения взаимодействия с административной панелью был реализован динамический просмотр деталей заявок и вопросов непосредственно из списка, без перехода на отдельную страницу. В административной панели также была настроена фильтрация и

сортировка для заявок и вопросов по различным критериям (дата, статус, тип клиента), а также поиск по ключевым полям (ФИО, телефон, текст запроса), что значительно повысило эффективность работы с архивами.

#### 8.4 Фронтенд-разработчик (Кудря А.С)

В ходе семестровой работы был полностью разработан клиентский интерфейс сайта с нуля, с использованием фреймворка Next.js и языка TypeScript.

Архитектура проекта построена на App Router, что позволило организовать маршруты и страницы в модульной и масштабируемой структуре. Все компоненты и стили созданы вручную, с применением CSS Modules для изоляции и удобства работы со стилями.

Реализованы ключевые разделы сайта: главная страница, информация о специалисте, услуги для физических и юридических лиц, статьи и блок с часто задаваемыми вопросами. Создана модальная форма обратной связи с возможностью отправки данных на сервер, отображаемая через портал.

Для контроля качества кода применялись строгая типизация, ESLint и настройка среды через конфигурационные файлы. Интерфейс адаптирован под разные разрешения экрана — 1440, 1024 и 768 пикселей. Все элементы протестированы вручную.

## ЗАКЛЮЧЕНИЕ

В результате выполнения проекта был создан персональный сайт для организационного психолога с системой администрирования, полностью соответствующий изначально сформулированным требованиям. Сайт предоставляет все ключевые функции: представление информации о специалисте и услугах, формы записи и обратной связи, блог с статьями, а также защищённую административную панель. Интеграция с Telegram обеспечивает своевременное информирование специалиста о новых заявках и вопросах, что существенно улучшает взаимодействие с клиентами.

На всех этапах разработки команда придерживалась гибкой методологии Agile, что позволило своевременно адаптировать продукт под пожелания заказчицы и быстро реагировать на возникающие проблемы. Благодаря использованию современных фреймворков и библиотек (Django, Next.js, TypeScript, CSS Modules), обеспечены высокая производительность, безопасность и масштабируемость решения.

По результатам тестирования были выявлены и устранены дефекты, связанные с отображением данных в административной панели и обработкой медиафайлов. Это повысило стабильность работы сайта и упростило взаимодействие администратора с системой. Также реализованы базовые меры защиты: HTTPS, валидация данных, защита от спама.

Сайт обладает адаптивной версткой и кроссбраузерностью, что делает его доступным на всех типах устройств. Пользовательский интерфейс прост и интуитивен, акцент сделан на удобство взаимодействия и визуальную доступность информации.

В дальнейшем возможны следующие направления развития продукта:

- добавление возможности оплаты услуг через сайт,
- расширение Telegram-интеграции (например, чат-бот для первичной консультации),
- внедрение аналитики посещаемости и поведения пользователей,

– реализация системы авторизации для постоянных клиентов с личным кабинетом.

Таким образом, проект можно признать успешно завершённым. Полученный результат обладает высоким качеством исполнения и высоким потенциалом для дальнейшего развития.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Django Software Foundation. Документация Django [Электронный ресурс]. – URL: <https://docs.djangoproject.com/> (дата обращения: 25.05.2025).
2. Vercel. Документация Next.js [Электронный ресурс]. – URL: <https://nextjs.org/docs> (дата обращения: 25.05.2025).
3. Meta. Документация React [Электронный ресурс]. – URL: <https://react.dev/> (дата обращения: 25.05.2025).
4. Telegram Messenger Inc. Telegram Bot API [Электронный ресурс]. – URL: <https://core.telegram.org/bots/api> (дата обращения: 25.05.2025).
5. Mypsy.org – сайт психологов [Электронный ресурс]. – URL: <https://myspy.org/> (дата обращения: 25.05.2025).
6. Ионычева, М. Персональный сайт психолога Marinaionycheva.ru [Электронный ресурс] / М. Ионычева. – URL: <https://marinaionycheva.ru/> (дата обращения: 25.05.2025).
7. Zigmund.online – сервис онлайн-психологов [Электронный ресурс]. – URL: <https://zigmund.online/> (дата обращения: 25.05.2025).
8. Lifshite.ru – сайт психолога Лифшица [Электронный ресурс]. – URL: <https://lifshite.ru/> (дата обращения: 25.05.2025).
9. Detok.org – информационный портал по психологии [Электронный ресурс]. – URL: <https://detok.org/> (дата обращения: 25.05.2025).
10. Материалы, предоставленные заказчиком: техническое задание, текстовое наполнение сайта, дизайн-референсы [непублик. источник].

## ПРИЛОЖЕНИЕ А

### Артефакты проекта

Google Disk. Все текстовые документы.

[https://drive.google.com/drive/folders/1k7FItRf\\_QuP4xzI\\_Snquf2ZUn3BVuFeK](https://drive.google.com/drive/folders/1k7FItRf_QuP4xzI_Snquf2ZUn3BVuFeK)

Таблица 1 – Задачи и функционал страниц сайта

Целевая аудитория	Задачи	Функционал	Страница
Потенциальные клиенты	Узнать общую информацию (о психологии, деятельность, видах услуг)	-	Главная страница
	Узнать про услугу: описание, детали, цена	-	Страница услуги
	Прочитать статью на интересующую тему	-	Страница статей
Заинтересованный пользователь	Задать волнующий вопрос	Отправка формы	Страница с формой создания вопроса
	Записаться на выбранную услугу	Отправка заявки	Страница с созданием заявки
Администратор	Получить вопрос от клиента	Получение уведомления + информации о новом вопросе в тг	Пул с вопросами?
	Получить заявку	Получение уведомления + информации о новой заявке в тг	Пул с заявками?
	Связаться с клиентом	Получение данных	Пул с заявками/вопросами??
	Добавить новую статью	Добавление статьи	Страница статей



Miro. Схемы и диаграммы.

[https://miro.com/app/board/uXjVIMyJwbI=/?share\\_link\\_id=767743007175](https://miro.com/app/board/uXjVIMyJwbI=/?share_link_id=767743007175)

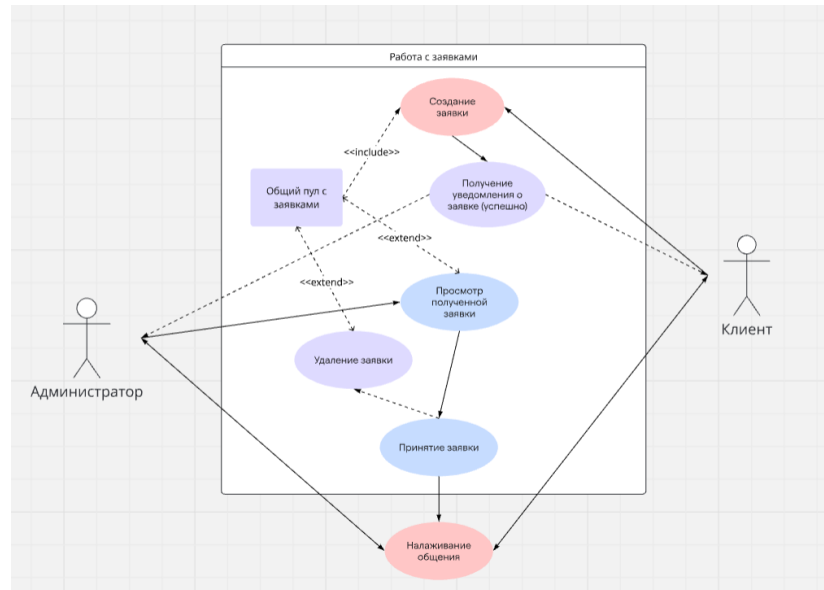


Рисунок 1 – Use Case Diagramm 1



Рисунок 2 – Use Case Diagramm 2

GitHub. Репозиторий проекта.

<https://github.com/mariaemel/project-org-psychologist>

Figma. Дизайн сайта.

<https://www.figma.com/design/WAFYwCleeGY6nlRJnkFczp/Untitled?node-id=0-1&t=EGXKDPzS6TBpyJYy-1>

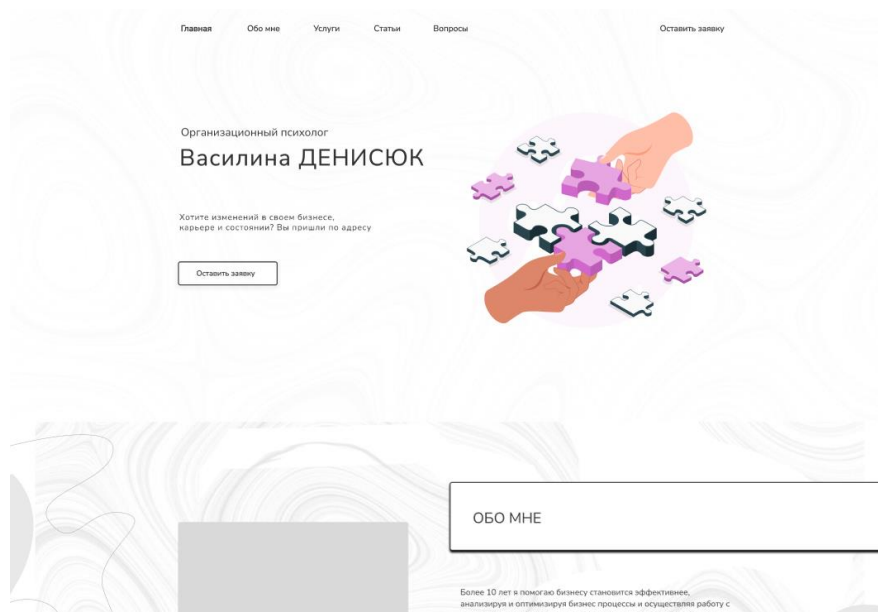


Рисунок 3 – Промежуточный дизайн

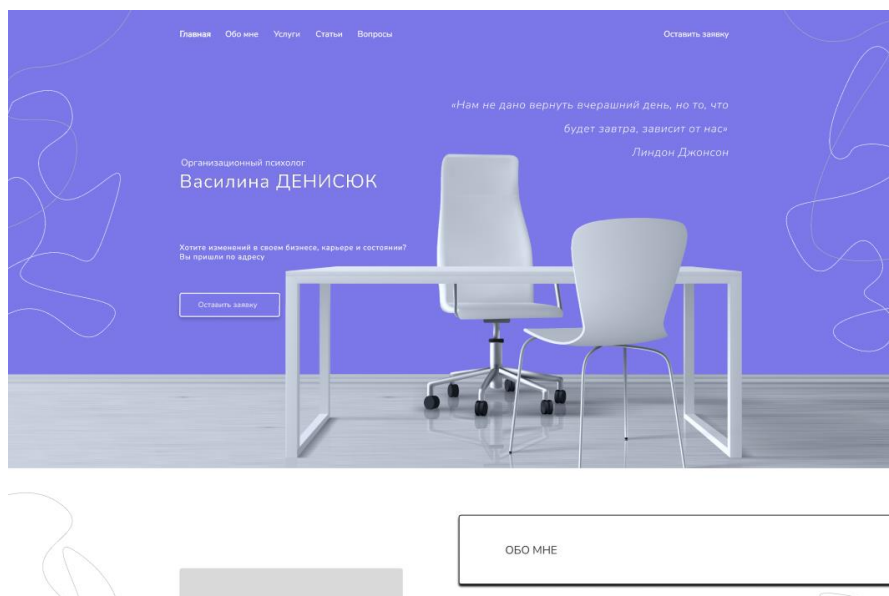


Рисунок 4 – Итоговый дизайн