

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-приложения для администрирования реферальной системы
международной школы программирования "Алгоритмика"(1)»
по дисциплине «Проектный практикум»

Заказчик: Звонова А.Н.

Куратор: Звонова А.Н.

Студенты команды _____

Назаров М.Е.

Киржацкий В.А.

Рукавишников А.С.

Коноплёв Р.Е.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1.1 Цель и задачи проекта	3
1.1.1 Основные задачи проекта:	3
1.2 Актуальность проекта	3
1.3 Область применения программного продукта.....	4
1.4 Ожидаемые результаты и планируемые достижения	4
2 Основная часть	6
2.1 Анализ требований заказчика и формирование бэклога	6
2.1.1 Выявление ключевых требований	6
2.1.2 Составление бэклога.....	7
2.2 Анализ аналогов и конкурентных решений.....	7
2.3 Архитектура программного продукта	9
2.3.1 Выбор технологического стека.....	10
2.3.2 Сценарий использования приложения	11
2.4 Отчет о процессе разработки приложения.....	13
2.4.1 Методология разработки.....	13
2.5 Отчеты членов команды.....	14
2.5.1 Отчет Назаров Михаил (Тимлид и Аналитик)	14
2.5.2 Отчет Киржацкий Владимир (Backend разработчик)	15
2.5.3 Отчет Коноплёв Роман (Дизайнер).....	16
2.5.4 Отчёт Рукавишников Александр (Frontend разработчик)	17
2.6 Доработка модулей.....	19
ЗАКЛЮЧЕНИЕ	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	22

ВВЕДЕНИЕ

1.1 Цель и задачи проекта

Целью проекта является разработка веб-сервиса реферальной системы для школы программирования "Алгоритмика", обеспечивающего эффективное привлечение новых пользователей, стимулирование рекомендаций и автоматизацию управления бонусами и скидками.

1.1.1 Основные задачи проекта:

- Разработка одноуровневой и многоуровневой системы реферальных вознаграждений.
- Реализация веб-интерфейса и Telegram-бота для взаимодействия с пользователями.
- Обеспечение администрирования курсов, акций и реферальных кампаний.
- Создание личного кабинета с визуализацией реферальной сети и историей начислений.

1.2 Актуальность проекта

Актуальность разработки обусловлена следующими факторами:

- Повышение конкуренции среди онлайн-школ требует инновационных инструментов привлечения клиентов.
- Существующие решения (Skillbox, GeekBrains и др.) предоставляют базовые одноуровневые системы без гибкости и визуализации структуры.
- Целевая аудитория проекта (родители учащихся) активно использует мессенджеры и склонна к «сарафанному радио», что делает реферальную модель особенно эффективной.
- Инструменты геймификации и аналитики способствуют повышению вовлеченности пользователей и увеличению продаж.

1.3 Область применения программного продукта

Программный продукт предназначен для:

- Частных и сетевых образовательных учреждений, использующих франшизу "Алгоритмика".
- Родителей учащихся, участвующих в реферальной программе и получающих бонусы.
- Администраторов и преподавателей для управления акциями, курсами и обращениями пользователей.
- Разработчиков и компаний, желающих масштабировать систему под свои нужды (в рамках мультикомпанийной архитектуры).
- Расширенного применения в сфере электронной коммерции и онлайн-сервисов (Post-MVP этап).

1.4 Ожидаемые результаты и планируемые достижения

По завершении проекта ожидается:

- Запуск MVP с полноценной поддержкой реферальной программы, Telegram-бота и веб-интерфейса.
- Повышение конверсии за счет гибкой структуры вознаграждений и персонализированных акций.
- Снижение затрат на маркетинг за счет автоматизации привлечения новых клиентов.
- Повышение лояльности аудитории благодаря геймификации и визуализации прогресса.
- Расширение системы до мультикомпанийного решения для подключения внешних партнеров.
- Масштабирование проекта при достижении 1000+ пользователей, внедрение персонализированной аналитики и рекомендаций.

После завершения этапа разработки всех модулей будет проведено масштабное тестирование с последующей доработкой продукта на основе обратной связи от пользователей.

2 Основная часть

2.1 Анализ требований заказчика и формирование бэклога

2.1.1 Выявление ключевых требований

На предварительном этапе разработки был проведён анализ требований заказчика — компании «Алгоритмика», а также ключевых стейкхолдеров проекта: администраторов учебного центра, родителей учащихся и преподавателей. Целью являлось создание масштабируемой, удобной и гибкой реферальной системы, способной:

- стимулировать привлечение новых клиентов через рекомендации,
- автоматизировать начисление бонусов и отображение реферальной структуры,
- предоставлять пользователю актуальную информацию о курсах, акциях и статусе своей реферальной сети.

На основе анализа были выделены следующие основные требования:

- реализация реферальной системы с поддержкой до 3 уровней вложенности,
- возможность генерации реферальных ссылок и QR-кодов на курсы и рекламные акции,
- создание визуального отображения структуры рефералов,
- интеграция с Telegram-ботом для регистрации, генерации ссылок, получения уведомлений и просмотра статистики,
- поддержка личного кабинета пользователя с историей начислений,
- разделы для управления курсами, акциями и пользователями в административной панели,
- наличие аналитики по привлечённым пользователям и эффективности кампаний.

Дополнительные пожелания от целевой аудитории:

- простой и интуитивный интерфейс, адаптированный под смартфоны и планшеты,

- гибкие настройки уведомлений (email, Telegram, push-уведомления),
- возможность персонализации реферальных предложений.

2.1.2 Составление бэклога

На основе выявленных требований был сформирован бэклог проекта, включающий следующие этапы и задачи:

- разработка и согласование технического задания (ТЗ) на MVP,
- проектирование интерфейсов веб-приложения и Telegram-бота (UX/UI дизайн),
- построение архитектуры базы данных с поддержкой многоуровневой структуры рефералов и кампаний,
- реализация backend-части на C#,
- разработка frontend-интерфейса на React19 + TypeScript,
- внедрение Telegram-бота с поддержкой ключевых функций: регистрация, генерация ссылок, статистика,
- проведение модульного и интеграционного тестирования всех компонентов,
- подключение pilotных филиалов "Алгоритмики" и сбор обратной связи,
- итеративная доработка системы с учётом отзывов пользователей и заказчика.

2.2 Анализ аналогов и конкурентных решений

Проведённый бенчмаркинг показал, что большинство онлайн-школ и образовательных платформ используют однотипные и упрощённые реферальные механизмы, часто ограниченные одним уровнем рекомендаций и базовыми бонусами.

Анализ проведён по следующим ключевым критериям: структура системы, гибкость ссылок, наличие Telegram-бота, визуализация сети и возможность продвижения акций (рисунок 1).

Критерий	Geekbrains	Skillbox	SkillFactor у	Хекслет	Stepik	Нетология	Coursera	Udemy	Наш проект для "Алгоритмики"
Уровни системы	1	1	1	1	1	1	1	1	1-3
Размер бонуса	до 10%	до 15%	до 10%	до 10%	до 15%	до 10%	варьир.	варьир.	сумма/5-15% (гибкий)
Платформы	Сайт+бот(огранич.)	Сайт+бот(огранич.)	Сайт	Сайт	Сайт	Сайт	Сайт	Сайт	Сайт+полноценный Tg bot
Привязка к акциям	Нет	Да*	Нет	Нет	Нет	Нет	Нет	Нет	Да(курсы/акции+гибкие настройки)
Визуализация	Список	Список	Список	Список	Список	Список	-	-	Интерактивный вывод
Геймификация	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Задания
Аналитика	Базовая	Базовая	Базовая	Базовая	Базовая	Базовая	-	-	Расширенная
Уведомления	Email + tg	Email + tg	Email + tg	Email	Email	Email	Email	Email	Email+tg
Регистрация и авторизация	Сайт	Сайт	Сайт	Сайт	Сайт	Сайт	Сайт	Сайт	Tg bot + Сайт

Рисунок 1 – анализ конкурентов

На основе анализа конкурентов были выявлены следующие особенности:

- Большинство платформ (Skillbox, Хекслет, Stepik) предлагают только одноуровневую систему с фиксированным бонусом 10–15%.
- Telegram-боты на большинстве платформ имеют ограниченный функционал, сводящийся к ссылкам на сайт и каналам новостей.
- Визуализация реферальной структуры и прогноза доходов отсутствует у всех исследуемых аналогов.
- Гибкие настройки и продвижение акций через реферальные ссылки не реализованы.
- Геймификация (мотивационные механизмы) встречается крайне редко и, как правило, в упрощённом виде.

Таким образом, существующие решения не в полной мере соответствуют современным требованиям пользователей и маркетологов, желающих использовать реферальные системы как инструмент для комплексного продвижения и удержания клиентов.

В отличие от конкурентов, наш проект предлагает расширенные возможности:

- гибкая многоуровневая структура,

- расширенный Telegram-бот с функциями регистрации, генерации ссылок, статистики и уведомлений,
- продвинутые инструменты визуализации и аналитики,
- поддержка акций и геймификации.

Это позволяет говорить о значительном конкурентном преимуществе системы и её потенциале как маркетингового инструмента.

2.3 Архитектура программного продукта

Разрабатываемое веб-приложение реферальной системы «Алгоритмика» построено по многоуровневой архитектуре, включающей frontend, backend, базу данных и Telegram-бот. Такая структура обеспечивает модульность, масштабируемость и удобство сопровождения проекта.

Приложение разделено на два основных интерфейсных уровня:

- Административная панель, предназначенная для управления пользователями, курсами, акциями, выплатами и статистикой;
- Пользовательский кабинет, ориентированный на родителей учащихся, с функциями генерации реферальных ссылок, просмотра рефералов и получения вознаграждений.

Проект также включает инфраструктурные решения:

- Развёртывание с использованием Docker Compose,
- Использование S3-хранилища для загрузки и хранения медиафайлов,
- Балансировка и маршрутизация через Nginx.

Архитектура бэкенда следует трёхслойной модели:

1. Слой представления (контроллеры),
2. Слой бизнес-логики,
3. Слой доступа к данным (модели, репозитории).

Системой предусмотрены ролевые ограничения: админ, суперадмин, преподаватель и пользователь, что обеспечивает безопасное разграничение доступа к данным и функциям.

2.3.1 Выбор технологического стека

В ходе проектирования и реализации веб-приложения были выбраны современные и широко используемые технологии, соответствующие требованиям по производительности, удобству сопровождения и расширяемости.

Frontend:

- React 19 + TypeScript — основной фреймворк для пользовательского интерфейса;
- Vite — сборщик, обеспечивающий высокую скорость разработки и оптимизацию production-сборки;
- Nx Monorepo — архитектура проекта с разделением на приложения и общие библиотеки;
- Tailwind CSS + Hero UI — кастомизируемая UI-библиотека с модульной стилистикой;
- Redux Toolkit + Redux Toolkit Query — управление состоянием и кэширование запросов;
- Axios — для работы с API;
- ESLint, Prettier, Husky — инструменты обеспечения качества кода.

Frontend-приложение включает три модуля:

- apps/admin — административная панель;
- apps/profile-mobile — мобильный личный кабинет;
- apps/home-app — лендинг с формой обратной связи и системой марафонов.

Backend:

- ASP.NET Core (C#) — фреймворк для реализации API, обеспечивающий высокую производительность и безопасность;
- PostgreSQL — СУБД для хранения пользовательских данных, реферальной структуры, покупок и начислений;
- S3-хранилище — используется для хранения медиафайлов (например, изображений курсов или отзывов);

- Docker + Docker Compose — управление контейнерами и развертывание приложения;
 - Nginx — веб-сервер и балансировщик нагрузки.
- Telegram-бот:
- Взаимодействует с backend через REST API;
 - Позволяет пользователю генерировать реферальные ссылки, просматривать статистику и получать уведомления.

Обоснование выбора технологий:

Выбор ASP.NET Core и React обусловлен высокой производительностью, поддержкой большого сообщества, наличием готовых решений и широкими возможностями по масштабированию. Использование Docker и монорепозитория Nx позволило обеспечить удобную разработку и внедрение CI/CD.

Технологический стек и архитектура закладывают надёжную основу для последующего расширения проекта: подключения новых компаний, создания расширенной аналитики и персонализированных рекомендаций.

2.3.2 Сценарий использования приложения

Сценарий использования веб-приложения реферальной системы «Алгоритмика» охватывает взаимодействие как со стороны конечного пользователя (родителя учащегося), так и со стороны администратора учебного центра. Ниже приведены типовые пользовательские пути:

1. Пользовательская часть (родитель)

1. Регистрация

и авторизация

Пользователь (родитель) проходит регистрацию через Telegram-бот.

После — создаётся личный кабинет. Авторизация осуществляется с помощью JWT-токена.

2. Просмотр информации о курсах и акциях

Пользователь заходит в раздел «Каталог курсов», где видит доступные курсы и акции.

3. Создание реферальной ссылки или QR-кода
Через личный кабинет или Telegram-бот пользователь генерирует:
 - ссылку на конкретный курс или акцию,
 - универсальную ссылку для привлечения новых пользователей.
4. Распространение ссылки и привлечение рефералов
Ссылки публикуются в мессенджерах, социальных сетях или пересылаются друзьям. При переходе по ссылке новый пользователь регистрируется, и система автоматически закрепляет его как реферала.
5. Начисление бонусов и визуализация структуры
После покупки курса рефералом, в личном кабинете пользователя отображается начисленный бонус. Также пользователь может видеть:
 - уровень реферала (1, 2 или 3),
 - структуру своей реферальной сети,
 - общий доход и историю транзакций.
6. Получение уведомлений
Пользователь получает уведомления о новых начислениях, акциях, изменениях в структуре через Telegram или браузерные push-уведомления.

2. Административная часть (менеджер/преподаватель)

1. Авторизация в админ-панели
Администратор входит в систему через защищённый интерфейс, используя логин и пароль. Роль определяет доступ к разделам (например, суперадмин, преподаватель, модератор).
2. Управление пользователями и курсами
В панели администратора можно:
 - просматривать список пользователей и их активность;
 - создавать, редактировать и архивировать курсы и акции;
 - фильтровать пользователей различными способами.

3. Модерация отзывов и обратной связи

Администратор может просматривать и обрабатывать обращения, поступившие через форму обратной связи или Telegram-бот.

4. Контроль выплат и бонусов

Система отображает начисленные бонусы, статистику по кампаниям, позволяет подтверждать или отклонять выплаты пользователям.

5. Аналитика

Доступен модуль с графиками, таблицами и фильтрами, отражающий конверсию, доход по филиалам, популярность курсов и эффективность реферальных программ.

2.4 Отчет о процессе разработки приложения

2.4.1 Методология разработки

Использованная методология: Гибридная (Scrum + Kanban).

Scrum для итеративной разработки (спринты по 2 недели).

Kanban для визуализации задач (доска в Jira).

Ключевые этапы:

1. Подготовка:

- Сбор требований, создание бэклога, проектирование архитектуры.

2. Разработка:

- Реализация модулей по приоритетам в рамках разных версий приложения.

3. Тестирование:

- Промежуточные тесты после каждого релиза.

4. Внедрение:

- Пилотное тестирование на тестовой фирме и пользователях.

5. Доработка:

- Доработка функционала приложения после получения обратной связи.

Роли команды:

- Тимлид: координация, аналитика.
- Дизайнер: проектирование интерфейсов.
- Разработчики: бэкенд (C#), фронтенд (React).

Примерный план реализации продукта представлен на таблице 1.

таблица 1

Этап	Действия	Сроки
Анализ	Интервью с заказчиками, бенчмаркинг	2 недели
Прототипирование	Создание ТЗ, дизайн UI в Figma, согласование с заказчиком	2 недели
Бэкенд-разработка	Реализация API (C#), миграции PostgreSQL	8 недель
Фронтенд	Верстка (React), интеграция с API	8 недель
Доработка	Возможная доработка приложения после получения обратной связи от пользователей.	2 недели

2.5 Отчеты членов команды

2.5.1 Отчет Назаров Михаил (Тимлид и Аналитик)

На начальном этапе проекта был проведён комплексный анализ конкурентных решений, а также изучены потребности заказчика. В рамках Customer Development были организованы интервью с представителями

целевой аудитории, что позволило сформировать обоснованные требования к функциональности системы.

- На основании полученной информации:
- сформирована дорожная карта проекта (roadmap), определены ключевые этапы разработки,
 - подготовлено техническое задание (ТЗ) для дизайнеров и разработчиков, включающее как функциональные, так и нефункциональные требования,
 - проведены согласования с заказчиком для утверждения состава работ и сроков их реализации.

В ходе основной фазы разработки выполнялись следующие задачи:

- координация процесса проектирования пользовательского интерфейса (UI/UX), контроль соответствия дизайна требованиям,
- декомпозиция задач — разбиение общей функциональности на отдельные задачи и подзадачи в системе управления проектами (Jira),
- контроль выполнения — регулярный мониторинг прогресса, проведение код-ревью,
- организация тестирования — проверка реализованных модулей с привлечением реальных пользователей, сбор и анализ обратной связи для последующей доработки системы.

- Результаты проделанной работы:
- успешно проведён анализ требований и согласование архитектурных решений с заказчиком,
 - организован процесс разработки, обеспечен контроль за качеством дизайна и реализацией функциональности.

В дальнейшем планируется реализация дополнительных доработок с учётом новых требований, поступивших в ходе эксплуатации MVP.

2.5.2 Отчет Киржацкий Владимир (Backend разработчик)

В ходе работы над проектом было разработано API веб-приложения, реализующее функциональность реферальной системы.

В качестве серверной части был выбран фреймворк ASP.NET Core. В качестве базы данных использовался кластер PostgreSQL. Для хранения медиафайлов настроено подключение к S3-хранилищу.

Архитектурно приложение разделено на две части: административную панель и пользовательский профиль. Административная панель включает модули управления фирмами, пользователями, оплатами, курсами, обратной связью, администраторами и отзывами.

Пользовательская часть включает модули отображения информации о пользователе, доходах, покупках, структуре рефералов, доступных акциях и курсах.

Реализовано разграничение прав доступа по ролям: админ, суперадмин, учитель, пользователь.

Все сущности описаны в виде моделей. Реализована бизнес-логика и слой представления через контроллеры (REST API). Бэкенд спроектирован по трёхуровневой архитектуре: слой представления, слой логики, слой данных, что обеспечивает масштабируемость и модульность.

С помощью конфигурации Docker Compose была развернута инфраструктура проекта, включающая nginx, backend и frontend. Приложение успешно развернуто на сервере.

Особое внимание было уделено проработке архитектуры приложения. Сложность взаимодействия 19 сущностей уже на текущем этапе потребовала закладывания гибкости и возможности масштабирования, что критически важно для дальнейшего развития и внесения изменений без потери устойчивости системы

2.5.3 Отчет Коноплёт Роман (Дизайнер)

В течение учебного семестра были выполнены задачи по проектированию и разработке пользовательского интерфейса для веб-приложения реферальной системы.

Были реализованы следующие ключевые модули:

- Мобильная версия сайта — создан адаптивный интерфейс, ориентированный на пользователей смартфонов. Особое внимание уделено удобству навигации, компактности элементов и сохранению полной функциональности при ограниченной площади экрана. Разработаны основные пользовательские сценарии: регистрация, авторизация, просмотр рефералов и генерация персональных ссылок.
- Десктопная версия сайта — обеспечен полнофункциональный интерфейс для пользователей ПК с расширенной визуализацией данных. Проработана структура страниц, внедрены адаптивные таблицы, панели с метриками и блоки уведомлений.
- Административная панель — создан интерфейс для администраторов системы, включающий модули управления пользователями, статистикой, системой начислений и подтверждения выплат. Реализована логическая иерархия элементов, обеспечена доступность функций, а также применены методы визуальной интерпретации данных (графики, таблицы, фильтры).

Дизайн разрабатывался в среде Figma с соблюдением принципов визуальной иерархии, модульной сетки и контрастности. В проектировании использовались современные UX-паттерны, проводился аудит удобства взаимодействия на различных разрешениях экранов.

Наиболее сложной задачей стала разработка универсальной навигационной системы, одинаково удобной как для мобильных, так и для десктопных пользователей. Было реализовано адаптивное поведение меню и контента, потребовавшее нескольких итераций и тестирования прототипов.

2.5.4 Отчёт Рукавишников Александр (Frontend разработчик)

В рамках разработки веб-приложения на React с использованием Vite и архитектуры Nx Monorepo была выполнена комплексная работа по настройке окружения, созданию интерфейсов и обеспечению взаимодействия с серверной частью.

Настройка окружения и архитектуры
Была инициализирована структура Nx Monorepo, что позволило централизованно управлять несколькими приложениями и общими библиотеками. В качестве сборщика применён Vite, обеспечивший высокую скорость разработки и оптимизированную сборку для production.

Базовая структура проекта включает:

- apps/admin — административная панель с функционалом модерации,
- apps/profile-mobile — мобильная версия пользовательского кабинета,
- apps/home-app — главная страница с лендингом,
- libs — библиотека общих компонентов, утилит и хуков.

Разработка интерфейсов
В административной панели реализованы:

- CRUD-операции с пользовательскими данными через таблицы,
- фильтрация и пагинация для упрощения модерации контента.

На главной странице разработаны:

- лендинг с интерактивным калькулятором,
- система марафонов с заданиями,
- форма обратной связи с валидацией.

Интеграция с бэкендом
Настроено взаимодействие с REST API с использованием:

- Axios и Redux Toolkit Query — для работы с запросами и кешированием данных,
- FormData — для отправки файлов в формах.

Реализованы ключевые функции,

- авторизация с использованием JWT и защищённые маршруты,
- оптимизация запросов (дебаунс, предотвращение повторных запросов).

Используемый технологический стек

- React 19 + TypeScript,
- Vite + Nx Monorepo,
- UI: Tailwind CSS, Hero UI, кастомные компоненты,
- State management: Redux Toolkit,
- API: Axios, Redux Toolkit Query,
- Инструменты обеспечения качества: ESLint, Prettier, Husky,

Решённые проблемы

- Оптимизация медленных сборок в Nx за счёт кеширования и пересмотра зависимостей;
- Внедрение строгой типизации и унифицированного API-клиента для предотвращения конфликтов при взаимодействии с API;
- Вынос логики управления состоянием в отдельные хуки и библиотеки для повышения модульности и читаемости кода.

Итоги

разработки

Разработаны и интегрированы три основных модуля:

- административная панель;
- мобильный профиль пользователя;
- главная страница с лендингом.

Обеспечена стабильная интеграция всех интерфейсов с backend-сервисами.

Планы по доработке

- Улучшение анимации и доступности интерфейсов;
- Оптимизация загрузки компонентов с использованием lazy loading;
- Проведение рефакторинга для снижения итогового размера сборки

2.6 Доработка модулей

В ходе пост-релизного тестирования веб-приложения реферальной системы «Алгоритмика», проведённого представителями целевой аудитории

(администраторами филиалов и родителями пользователей), была зафиксирована необходимость лишь малых доработок – исправление багов и ошибок.

- | | | |
|------------|-------------|------------|
| Значимость | проведённых | изменений: |
|------------|-------------|------------|
- Уточнение пользовательских сценариев на основе реального фидбека позволило адаптировать модули к практическим задачам конечных пользователей.
 - Улучшение Telegram-бота повысило вовлечённость пользователей и сократило путь от рекомендации до действия.
 - Визуализация и гибкость реферальной структуры усилили восприятие «сетевого эффекта», тем самым стимулируя дальнейшее распространение программы.

Проведённые доработки отражают итеративный подход к разработке, при котором каждое обновление базируется на пользовательской обратной связи. Такой подход обеспечивает более точное соответствие системы как техническим, так и поведенческим ожиданиям целевой аудитории.

ЗАКЛЮЧЕНИЕ

В рамках реализации проекта была разработана и внедрена реферальная система для школы программирования «Алгоритмика», включающая веб-приложение, административную панель и Telegram-бот. Проект ориентирован на расширение клиентской базы за счёт привлечения новых пользователей через многоуровневую систему рекомендаций и гибкие механизмы мотивации.

На всех этапах — от анализа требований и конкурентной среды до архитектурного проектирования и создания интерфейсов — соблюдался подход, основанный на взаимодействии с целевой аудиторией и итеративном улучшении продукта. Благодаря этому удалось реализовать систему, которая:

- обладает высоким уровнем пользовательского удобства и визуальной наглядности,
- предлагает продвинутые возможности настройки реферальных акций,
- интегрирует аналитику, геймификацию и Telegram-бот для повышения вовлечённости,
- адаптирована как под десктопные, так и мобильные устройства.

Разработанная архитектура обеспечивает масштабируемость и техническую устойчивость, позволяя в дальнейшем расширять систему до мультикомпанийного формата с возможностью подключения внешних партнёрских организаций.

Проект успешно прошёл стадию MVP и может быть использован как в рамках существующих франшиз «Алгоритмика», так и адаптирован под другие образовательные или коммерческие структуры. Полученные в ходе работы навыки проектирования, разработки и тестирования сложных веб-систем, а также взаимодействия с реальными пользователями, подтверждают как прикладную значимость, так и образовательную ценность проекта

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Официальная документация ASP.NET Core. Microsoft Docs. — URL: <https://learn.microsoft.com/aspnet/core>
2. PostgreSQL Documentation. — URL: <https://www.postgresql.org/docs/>
3. Docker Documentation. — URL: <https://docs.docker.com/>
4. React Documentation. — URL: <https://react.dev/>
5. Redux Toolkit Documentation. — URL: <https://redux-toolkit.js.org/>
6. Vite — Next Generation Frontend Tooling. — URL: <https://vitejs.dev/>
7. Nx Monorepo Framework. — URL: <https://nx.dev/>
8. Telegram Bot API. — URL: <https://core.telegram.org/bots/api>
9. Tailwind CSS Documentation. — URL: <https://tailwindcss.com/docs>
10. Figma Community Guidelines on UX/UI Patterns. — URL: <https://www.figma.com/community>
11. Исследование рынка EdTech в России. Аналитический центр EdMarket, 2023.