

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 онлайн-генератора сертификатов для сайта учреждения»
по дисциплине «Проектный практикум»

Заказчик: Шишкина В.В.

Куратор: Шестеров М.А.

ученая степень, ученое звание, должность

Студенты команды _____

Алиев М.А.

Зарембо А.А.

Куртов Н.Е.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Основная часть	5
1.1 Работа участников команды.....	5
1.1.1 Зарембо Александр Александрович (аналитик)	5
1.1.2 Алиев Максим Амиршоевич (frontend-разработчик).....	5
1.1.3 Куртов Никита Евгеньевич (backend-разработчик).....	5
1.2 Разбор требований заказчика и пользователей, составление плана	6
1.3 Анализ и сопоставление аналогов.....	6
1.4 Архитектура продукта	7
1.5 Методология и процесс разработки	9
1.5.1 Сбор и анализ требований.....	9
1.5.2 Проектирование архитектуры.....	10
1.5.3 Разработка.....	10
1.6 Планирование деятельности и распределение задач участников	10
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	13

ВВЕДЕНИЕ

Проект направлен на создание веб-приложения, позволяющего пользователям автоматически генерировать именные сертификаты. Основной целью проекта является разработка удобного инструмента для автоматизации процесса создания сертификатов с возможностью ввода ФИО, выбора формата изображения и настройки внешнего вида шаблонов. Ключевые задачи проекта:

- 1) реализация интерфейса для настройки шаблонов (включая размер текста, положение, шрифт);
- 2) разработка формы для ввода данных пользователем;
- 3) обеспечение функции сохранения и изменения созданных шаблонов.

Актуальность проекта обусловлена необходимостью оптимизации ручного процесса подготовки сертификатов, который на сегодняшний день отнимает значительное количество времени у сотрудников и совмещён с высокой вероятностью допущения ошибок. В условиях активного проведения дистанционных мероприятий, обучающих курсов и различных конкурсов, автоматизация генерации сертификатов становится особенно важной для учреждений, которые работают с большим числом участников.

Продукт охватывает учреждения культуры, образования и другие организации, регулярно выдающие сертификаты своим клиентам и сотрудникам. В частности, целевой аудиторией проекта выступают Центральная библиотечная система (ЦБС), её сотрудники и клиенты, которым предоставляется возможность самостоятельно получить сертификат через сайт учреждения.

По завершении проекта ожидается внедрение полноценного веб-приложения, обеспечивающего следующие возможности:

- 1) создание, редактирование и сохранение шаблонов сертификатов;
- 2) автоматическую генерацию именных сертификатов на основе введённых данных;

- 3) удобный и интуитивно понятный интерфейс как для сотрудников учреждения, так и для конечных пользователей;
- 4) сокращение временных затрат и снижение количества ошибок при оформлении сертификатов.

Реализация проекта повысит эффективность работы учреждения и улучшит качество обслуживания пользователей.

1 Основная часть

1.1 Работа участников команды

1.1.1 Зарембо Александр Александрович (аналитик)

Занимался сбором и систематизацией требований заказчика, анализом целевой аудитории и конкурентных решений. На основе требований сформулировал план задач backlog(рисунок 1), а также участвовал в планировании архитектуры приложения. Курировал соответствие реализуемого функционала поставленным требованиям на всех этапах проекта и проводил тестирование интерфейсов.

1.1.2 Алиев Максим Амиршоевич (frontend-разработчик)

Разработал пользовательский интерфейс генератора. В частности, реализовал административную панель, в которой можно загрузить шаблон сертификата и разместить на нём текст (например, поле для ФИО), с возможностью изменения положения и размера шрифта. Также создал интерфейс для пользователей, где можно ввести своё имя и скачать готовый сертификат. Все страницы были разработаны с учётом отзывчивости и пользовательского удобства.

1.1.3 Куртов Никита Евгеньевич (backend-разработчик)

Создал серверную часть проекта, включающую работу с базой данных. Бэкенд обрабатывает загрузку шаблонов, сохранение и редактирование данных, выдаёт шаблоны по ID, а также позволяет получать первые N шаблонов. Была реализована базовая REST API для интеграции с фронтендом и подготовки к дальнейшей публикации на сайте учреждения. Также

предусмотрена архитектура для последующей поддержки экспорта в форматы PDF, JPG и PNG.

1.2 Разбор требований заказчика и пользователей, составление плана

Требования заказчика включали в себя следующие пункты:

- 1) Создание онлайн-генератора сертификатов;
- 2) Поддержка ввода ФИО пользователем;
- 3) Поддержка форматов PDF, JPG и PNG;
- 4) Удобство использования для посетителей сайта.

В свою очередь требования пользователя должны удовлетворять следующие пункты:

- 1) Возможность самостоятельно ввести имя;
- 2) Получение готового документа без регистрации и лишних действий;
- 3) Быстрая генерация и скачивание в нужном формате.

Для выполнений данных требований был разработан план задач (рисунок 1):

Задача	Ответственный	Дата										
		07.04	14.04	21.04	28.04	05.05	12.05	19.05	26.05	02.06		
Обсуждение проекта с заказчиком	Алиев Максим											
Обзор аналогов	Зарембо Александр											
Составление плана работ	Куртов Никита											
Создание дизайна	Алиев Максим											
Создание архитектуры backend	Куртов Никита											
Создание use-case	Зарембо Александр											
Создание функций сохранения шаблонов	Куртов Никита											
Создание функций генерации	Алиев Максим											
Демонстрация макетов заказчику	Зарембо Александр											
Реализация макетов в готовую верстку	Алиев Максим											
Создание API	Куртов Никита											
Деплой backend	Куртов Никита											
Использовать API	Алиев Максим											
Подготовка к защите и демонстрация продукта	Зарембо Александр											

Рисунок 1 – План задач (Backlog)

1.3 Анализ и сопоставление аналогов

Были изучены и сравнены существующие решения: Flyvi, SUPA, GroupDocs, Certifier (рисунок 2). Наиболее близким по целям оказался сервис Certifier, предлагающий массовую генерацию, визуальный редактор и экспорт в различные форматы. Однако он требует платного доступа и сложен в использовании для простых пользователей.

Разрабатываемый генератор отличается от аналогов, включая в себя следующее:

- 1) простота в освоении;
- 2) интеграция напрямую в сайт библиотеки;
- 3) поддержка загрузки кастомных шаблонов.

Таким образом, проект закрывает нишу между простыми сервисами вроде Flyvi и профессиональными решениями как Certifier, предлагая удобные функции без избыточной сложности.

Критерий / Сервис	Flyvi	SUPA	GroupDocs	Certifier
Простота использования	9	8	5	7
Поддержка форматов	4	10	5	10
Визуальный редактор	6	10	0	9
Массовая генерация	8	2	2	10
Возможность интеграции	4	3	1	10

Рисунок 2 – Сравнение аналогов

1.4 Архитектура продукта

Фронтенд отвечает за отображение интерфейса и взаимодействие с пользователем. С его помощью можно загружать изображения, просматривать информацию о них, редактировать и удалять данные. Все действия пользователя на фронтенде отправляются на сервер с помощью HTTP-запросов, например POST, GET, PUT и DELETE.

Бэкенд реализует REST API и принимает запросы от фронта. При получении запроса он обрабатывает данные, взаимодействует с базой данных и возвращает ответ. В проекте предусмотрено четыре основных эндпоинта. Первый — GET /api/imageupload/{id} — используется для получения информации об изображении по его идентификатору. Второй — POST /api/imageupload/imageupload — позволяет загрузить новое изображение на сервер вместе с его метаданными. Третий — PUT /api/imageupload/{id} — обновляет существующую запись в базе, например, редактирует описание или название изображения. Четвёртый — DELETE /api/imageupload/{id} — удаляет изображение и его данные из системы.

База данных служит для хранения всей информации, связанной с изображениями. Это может быть как сама картинка (например, в бинарном формате), так и сопутствующие данные: имя файла, дата загрузки, описание и путь к файлу, если изображения сохраняются в файловой системе. Сервер взаимодействует с базой при каждом запросе: при создании записей, их чтении, обновлении и удалении.

Вся архитектура (рисунок 3) строится на принципе последовательного взаимодействия. Когда пользователь загружает изображение, фронтенд отправляет его на бэкенд, который сохраняет данные и передаёт их в базу. При запросе изображения всё происходит в обратном порядке — сначала данные извлекаются из базы, затем отправляются клиенту. Такой подход обеспечивает чёткое разделение ответственности между компонентами и позволяет масштабировать проект в будущем.

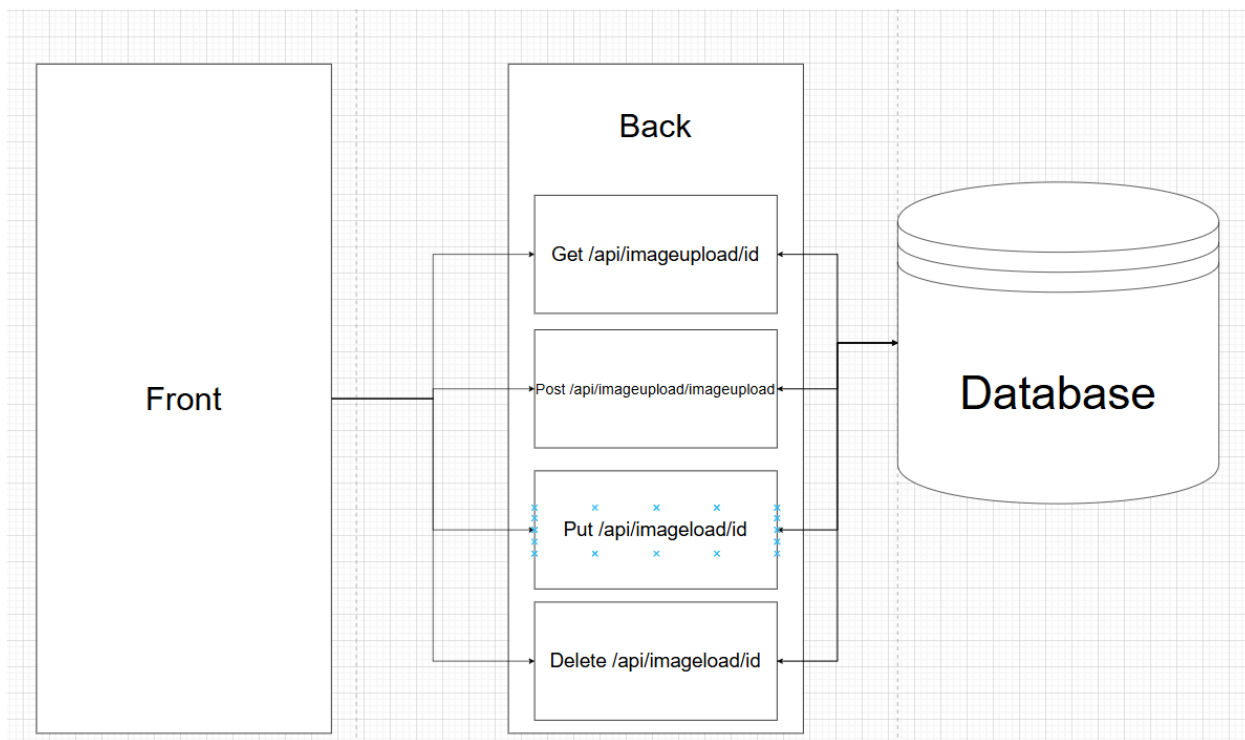


Рисунок 3 – Архитектура

1.5 Методология и процесс разработки

В рамках применялась каскадная модель разработки программного обеспечения Waterfall[1], предполагающая последовательное прохождение всех стадий жизненного цикла разработки. Такой подход был выбран в связи с тем, что требования заказчика были чётко сформулированы на этапе старта проекта и не предполагали значительных изменений в процессе реализации.

Процесс разработки включал следующие этапы:

1.5.1 Сбор и анализ требований

На этом этапе были собраны и структурированы требования заказчика и конечных пользователей, а также сформулированы ключевые функции системы. Было установлено, что для полного соответствия требованиям заказчика нам необходимо сделать следующее:

- 1) Создание онлайн-генератора сертификатов;

- 2) Поддержка ввода ФИО пользователем;
- 3) Поддержка форматов PDF, JPG и PNG;
- 4) Сохранение готовых шаблонов
- 5) Возможность редактирования сохранённых шаблонов
- 6) Возможность удаления шаблонов
- 7) Просмотр всех шаблонов

1.5.2 Проектирование архитектуры

Был выбран стек технологий. Мы выбрали .Net Core для backend-приложения, из-за его надежности и кроссплатформенности, PostgreSQL из-за удобной расширяемости и стабильности, React + TypeScript из-за гибкого и масштабируемого UI. Была спроектирована архитектура продукта, включая структуру базы данных, взаимодействие клиентской и серверной части и основные сценарии использования. Дизайнером был разработан макет в Figma[2].

1.5.3 Разработка

Frontend-разработчик реализовал интерфейсы администратора и пользователя, включая визуальное редактирование шаблонов. Backend-разработчик создал серверную логику, обеспечил хранение, обработку данных и поддержку работы с шаблонами.

1.6 Планирование деятельности и распределение задач участников

Процесс разработки был построен так, чтобы заказчик получил продукт, который полностью соответствует его требованиям. Для этого между разработчиками и заказчиком в роли связующего звена находился аналитик.

Он демонстрировал заказчику имеющиеся наработки и решения, получал обратную связь и давал указания разработчикам. Backend-разработчик отвечал за создание базы данных, а также за разработку backend-приложения. Frontend-разработчик был ответственным за создания визуального стиля сайта и его разработку. Передача наработок между разработчиками происходила с помощью Git[3] и платформы GitHub[4]. Таким образом у каждого участника был свой список задач. Список задач находится в приложении.

ЗАКЛЮЧЕНИЕ

Разработка онлайн-генератора сертификатов была направлена на решение актуальной проблемы — оптимизация процесса выдачи именных документов пользователям учреждений. В результате удалось реализовать продукт, который в целом соответствует поставленным требованиям заказчика. Генератор включает в себя: возможность добавления ФИО, визуальное редактирование шаблона на стороне администратора, генерацию уникальной ссылки на сертификат и последующее скачивание документа пользователем. Несмотря на то, что на момент завершения текущего этапа разработки поддержка всех форматов (JPG, PDF, PNG) ещё находится в процессе реализации, генератор работает и система готова к дальнейшему расширению.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Waterfall или Agile, Scrum или Kanban: что выбрать / [Электронный ресурс] // Хабр : [сайт]. — URL: <https://habr.com/ru/companies/kaiten/articles/906006/> (дата обращения: 26.05.2025).
2. Figma / [Электронный ресурс] // Figma : [сайт]. — URL: www.figma.com (дата обращения: 26.05.2025).
3. Git / [Электронный ресурс] // Git : [сайт]. — URL: <https://git-scm.com> (дата обращения: 26.05.2025).
4. GitHub / [Электронный ресурс] // GitHub : [сайт]. — URL: <https://github.com> (дата обращения: 26.05.2025).