

Федеральное государственное автономное образовательное учреждение высшего
образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра «Информатика и информационные технологии»

Направление подготовки/ специальность: 09.03.02 - Информационные системы и
технологии

ОТЧЕТ

по проектной практике

Студент: Голданова Любовь Владимировна. Группа: 241-334

Место прохождения практики: Московский Политех, кафедра «Информатика и
информационные технологии»

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Рябчикова Анна Валерьевна

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ	4
1.1 Наименование проекта.....	4
1.2 Описание проекта	4
2. ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ.....	6
2.1 Наименование заказчика	6
2.2 Организационная структура.....	6
2.3 Описание деятельности предприятия	6
3. ОПИСАНИЕ ЗАДАНИЯ ПО ПРОЕКТНОЙ ПРАКТИКЕ	8
3.1 Базовая часть задания	8
3.2 Вариативная часть задания.....	9
4. ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ ПО ПРАКТИКЕ	10
4.1 Результат выполнения базовой части	10
4.1.1 Используемые технологии.....	10
4.1.2 Разработка сайта	11
4.2 Результат выполнения вариативной части	20
4.2.1 Используемые технологии.....	20
4.2.2 Разработка блокировщика	21
4.3 Ссылки на GitHub	25
ЗАКЛЮЧЕНИЕ	26
СПИСОК ЛИТЕРАТУРЫ.....	27

ВВЕДЕНИЕ

Современные технологии виртуальной реальности (VR) и искусственного интеллекта (ИИ) стремительно меняют цифровой ландшафт, открывая новые горизонты для бизнеса, творчества и социального взаимодействия.

В последнее время люди все чаще стали говорить о метавселенных: информацию о виртуальных мирах публикуют в новостях, выходят игры и фильмы, повествующие о цифровой реальности.

С появлением интернета, человечество стало двигаться в сторону преобразования этой сети к более удобным социальным канонам. Сначала простой текст перерос в изображения, после в видео и, наконец, в живые трансляции. Но общение в сети, как и прежде, не выходит за рамки компьютерных экранов.

Суть метавселенных в том, что они и являются следующим шагом в развитии интернет технологий и взаимодействия внутри сети. В метавселенной, при использовании разных сетевых услуг и сервисов, у каждого пользователя будет личный анимированный 3D аватар, наподобие персонажа компьютерной игры, но абсолютно уникальный. Благодаря метавселенной, взаимодействие людей в интернете будет намного разнообразнее и богаче, ощущения станут более реальными.

У пользователей возрастает необходимость иметь уникальные способы самовыражения — реалистичные аватары отвечают этой потребности. Современная индустрия виртуальной реальности (VR) и игр стремительно развивается, предоставляя пользователям всё больше возможностей для взаимодействия в цифровых мирах.

Технологии открытого исходного кода (нейронные сети Hunyuan, Trellis) делают возможным быстрый старт бизнеса без колоссальных затрат. Texel предлагает поддержку и менторство, помогая преодолеть барьеры входа в высокотехнологичную сферу. Это особенно актуально для начинающих предпринимателей и студентов, желающих углубить компетенции в AI и VR.

1. ОБЩАЯ ИНФОРМАЦИЯ О ПРОЕКТЕ

1.1 Наименование проекта

Проект - Собственный бизнес в партнёрстве с «Texel». Аватары для метавселенных, VR и игр.

1.2 Описание проекта

Суть проекта

Проект представляет собой сервис, позволяющий пользователям быстро и просто создавать реалистичные 3D-аватары, совместимые с популярными VR-платформами (VRChat, Roblox, Horizon Worlds) и игровыми движками. В рамках проекта предусмотрено обучение и поддержка предпринимателей, желающих развивать собственные стартапы в области VR и AI.

Цели проекта

- 1) Создать интуитивно понятный сервис для генерации высококачественных 3D-аватаров с возможностью их экспорта в VR и игровые движки.
- 2) Обучить участников основам бизнеса в VR – от разработки MVP до поиска инвестиций.
- 3) Обеспечить практическое обучение технологиям нейронных сетей (Hunyuan, Trellis) и интеграции с игровыми платформами.
- 4) Развивать навыки работы с AI и обучать эффективному использованию ChatGPT для бизнеса.
- 5) Предоставить участникам возможности карьерного роста в Texel через стажировки, практику и трудоустройство.

Задачи проекта

- 1) Исследовательская часть

- Анализ возможностей Hunyuan, Trellis и других AI-библиотек для создания фотореалистичных 3D-аватаров.
- Изучение методов риггинга и анимации для совместимости с Unity и Unreal Engine.

2) Прототипирование

- Разработка пользовательского интерфейса сервиса (загрузка фото, генерация аватара, просмотр результата).
- Создание демонстрационных 3D-аватаров для тестирования и презентаций.

3) Рыночное тестирование

- Работа с ранними пользователями из VR и игровых сообществ.
- Взаимодействие с инфлюенсерами и стримерами для продвижения проекта.

4) Масштабирование

- Подготовка инвестиционных презентаций, поиск грантов и партнёров.
- Продвижение через СМИ и разработка кобрендинговых инициатив.

2. ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИИ

2.1 Наименование заказчика

Заказчик - компания ООО «ТЕКСЕЛ», зарегистрированная в Москве 12 августа 2014 г.

2.2 Организационная структура



Рисунок 1 – Структура предприятия

2.3 Описание деятельности предприятия

«ТЕКСЕЛ» — производитель профессиональных 3D-сканеров и разработчик программного обеспечения на базе искусственного интеллекта для создания цифровых аватаров.

Основные направления деятельности

1) 3D-сканирование человека - Компания входит в число лидеров мирового рынка по производству полноростовых 3D-сканеров. Эти устройства снимают внешние параметры тела (размеры, пропорции) с высокой точностью и создают цифровые модели человека за считанные минуты. Они используются в более чем 30 странах в сферах моды, развлечений и образования.

2) Виртуальная примерка и подбор размеров - «ТЕКСЕЛ» разработала виджет, позволяющий покупателям виртуально примерять одежду онлайн по одной фотографии. Эта технология активно внедряется в интернет-магазинах, улучшая пользовательский опыт и снижая количество возвратов.

3) Интеграция искусственного интеллекта - Компания активно использует искусственный интеллект в своих продуктах и внутренних процессах, включая стажировки и найм сотрудников, менеджмент, продажи, техподдержку, научные исследования, разработку, маркетинг и юридические аспекты.

4) Научные исследования и разработки - «ТЕКСЕЛ» ведет научные исследования и разработки в области естественных и технических наук. Они:

- Улучшают качество захвата 3D-изображений.
- Разрабатывают новые методы обработки больших данных 3D-сканов для ускорения их рендеринга и анализа.
- Исследуют поведение тканей и одежды на теле для улучшения точности виртуальной примерки.
- Ведут научно-исследовательские работы в области 3D-сканирования, компьютерного зрения и машинного обучения.

Компания сотрудничает с научными институтами и технологическими компаниями в рамках совместных разработок, а также получает гранты и поддержку от Фонда Сколково и правительства Москвы.

5) Применение технологий

Решения компании находят применение в различных сферах:

- Ритейл: виртуальные примерочные в торговых центрах, таких как ТРЦ «Планета» в Уфе.
- Фитнес и здоровье: оценка прогресса в фитнесе с использованием 3D-сканирования.
- Косметология: прогнозирование и рекомендации в косметологических услугах.
- Развлечения: создание цифровых аватаров для игр и виртуальной реальности.

3. ОПИСАНИЕ ЗАДАНИЯ ПО ПРОЕКТНОЙ ПРАКТИКЕ

3.1 Базовая часть задания

1) Настройка Git и репозитория:

- Создание личного репозитория на GitHub или GitVerse.
- Изучение базовых команд Git: клонирование, коммит, пуш и создание веток.
- Фиксация изменения с осмысленными сообщениями к коммитам.

2) Написание документов в Markdown:

- Оформление материалов проекта (описание, журнал прогресса и др.) в формате Markdown.
- Изучение синтаксиса Markdown и подготовка документов.

3) Создание статического веб-сайта:

- Использование технологий HTML и CSS для создания сайта.
- Создание сайта об основном проекте по дисциплине «Проектная деятельность». Оформление и наполнение сайта должны быть уникальными более, чем на 50%.
- Оформление страниц сайта с использованием графических материалов (фотографий, схем, диаграмм, иллюстраций) и других медиа элементов (видео)

4) Сайт должен включать:

- Домашнюю страницу с аннотацией проекта.
- Страницу «О проекте» с описанием проекта.
- Страницу или раздел «Участники» с описанием личного вклада каждого участника группы в проект по «Проектной деятельности».
- Страницу или раздел «Журнал» с минимум тремя постами (новостями, блоками) о прогрессе работы.
- Страницу «Ресурсы» со ссылками на полезные материалы (ссылки на организацию-партнёра, сайты и статьи, позволяющие лучше понять суть проекта).

5) Взаимодействие с организацией-партнёром:

- Организация взаимодействия с партнёрской организацией (визит, онлайн-встреча или стажировка).
- Участие в профильных мероприятиях по тематике проекта и профилю организации-партнёра (конференции, выставки, митапы, семинары, хакатоны).

3.2 Вариативная часть задания

Необходимо разработать собственный блокировщик рекламы для браузера на JavaScript.

1) Провести исследование: изучить, как создать выбранную технологию с нуля, воспроизвести практическую часть.

2) Создать подробное описание в формате Markdown, включающее:

- Последовательность действий по исследованию предметной области и созданию технологии.
- Техническое руководство по созданию этой технологии, ориентированное на начинающих. Оно должно включать пошаговые инструкции, примеры кода, иллюстрации (картинки, диаграммы, схемы) в количестве от 3 до 10 штук, вставленные в текст для наглядности.
- Поместить результаты исследования и руководства в общий Git-репозиторий.
- Сделать модификацию проекта согласно полученным знаниям и навыкам. Описать в технической документации модификации.
- Задokumentировать проект в репозитории.

4. ОПИСАНИЕ ДОСТИГНУТЫХ РЕЗУЛЬТАТОВ ПО ПРОЕКТНОЙ ПРАКТИКЕ

4.1 Результат выполнения базовой части

В ходе выполнения задания был разработан веб-сайт на языке HTML и CSS, описывающий работу по предмету «Проектная деятельность».

4.1.1 Используемые технологии

HTML (HyperText Markup Language) и CSS (Cascading Style Sheets) - это основы веб-разработки. HTML служит для создания структуры содержимого веб-сайта, в то время как CSS отвечает за визуальное оформление. Оба языка совместно позволяют создавать уникальные и привлекательные веб-сайты.

Преимущества HTML и CSS

1. Раздельность: HTML и CSS раздельны друг от друга, что упрощает поддержку и модификацию веб-сайта. Изменения в стиле оформления, которые вносятся с помощью CSS, не требуют изменения HTML-кода сайта.

2. Универсальность: HTML и CSS широко используются по всему интернету и поддерживаются всеми современными браузерами. Это означает, что веб-сайты, созданные с помощью HTML и CSS, будут хорошо отображаться на разных устройствах и платформах.

3. Гибкость: с помощью CSS можно легко изменять стилизацию веб-сайта. Это позволяет создавать уникальные и привлекательные дизайны, а также адаптировать сайт под различные разрешения экрана.

4. Улучшенная доступность: HTML и CSS обеспечивают возможность создания веб-сайтов, которые доступны для пользователей с ограниченными возможностями. Они поддерживают использование различных атрибутов и свойств для улучшения опыта взаимодействия и навигации.

Именно поэтому для написания сайта были выбраны эти языки

4.1.2 Разработка сайта

1) Общий стиль сайта

Так как темой проекта является Аватары для метавселенных, VR и игр, стиль сайта был выбран футуристический, неоново-киберпанковский. Он идеально подходит, потому что:

- Он отражает цифровой мир и виртуальность. Темный фон с градиентами имитирует космос, создавая ощущение бесконечности, как в метавселенных. Неоновые элементы ассоциируются с цифровыми интерфейсами, голограммами и виртуальной реальностью. Анимированные частицы напоминают пиксели, коды или цифровую материю, что перекликается с идеей симуляции.
- Связывает с гейм-дизайном и киберкультурой - Эффекты свечения напоминают HUD (интерфейсы дополненной реальности) из sci-fi и VR-игр.
- Динамика и интерактивность придают ощущение VR
- Анимации (плавающие элементы, пульсация, hover-эффекты) создают эффект "живого" цифрового пространства, что соответствует интерактивности метавселенных.
- 3D-эффекты и трансформации (кнопки, логотип) визуально передают идею объемных аватаров и виртуальных миров.

Код стиля представлен в листинге 1.

Листинг 1 – стиль сайта

```
html, body {
    height: 100%;
    margin: 0;
}
body {
    display: flex;
    flex-direction: column;
    background:
        radial-gradient(circle at 20% 30%, rgba(70, 130, 180, 0.08) 0%, transparent 40%),
        radial-gradient(circle at 80% 70%, rgba(255, 105, 180, 0.07) 0%, transparent 45%),
        linear-gradient(135deg, #121a2e 0%, #2a1f40 100%);
    position: relative;
    background-attachment: fixed;
```

```

}
.particle {
  position: absolute;
  border-radius: 50%;
  background: linear-gradient(45deg, #6e44ff, #ff44a8);
  filter: blur(5px);
  opacity: 0.3;
  animation: float 15s infinite linear;
}
@keyframes float {
  0%, 100% { transform: translate(0, 0); }
  25% { transform: translate(50px, -30px); }
  50% { transform: translate(20px, -50px); }
  75% { transform: translate(-30px, -20px); }
}
:root {
  --primary: #6e44ff;
  --secondary: #b892ff;
  --accent: #ff44a8;
  --dark: #0f0a20;
  --light: #f0eaff;
  --neon-glow: 0 0 10px var(--primary), 0 0 20px var(--secondary), 0 0 30px var(--
accent);
}
header {
  font-family: 'Orbitron', sans-serif;
  background-color: var(--dark);
  color: white;
  height: 130px;
  overflow: hidden;
  position: fixed;
  width: 100%;
  z-index: 1;
}
.floating-avatars {
  position: absolute;
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
  z-index: -1;
}
.parallax-bg {
  position: absolute;
  width: 120%;
  height: 120%;
  top: -10%;
  left: -10%;
  background:
    radial-gradient(circle at 30% 50%, rgba(110, 68, 255, 0.15) 0%, transparent 40%),
    radial-gradient(circle at 70% 30%, rgba(184, 146, 255, 0.1) 0%, transparent 45%),

```

```

        radial-gradient(circle at 50% 80%, rgba(255, 68, 168, 0.1) 0%, transparent 40%);
background-size: 200% 200%;
will-change: transform;
z-index: -2;
}
.header-container {
display: flex;
align-items: center;
padding: 15px;
text-align: center;
position: relative;
z-index: 1;
}
.header-logo {
font-size: 4rem;
font-weight: 700;
background: linear-gradient(45deg, var(--primary), var(--accent), var(--secondary));
background-clip: text;
color: transparent;
text-shadow: var(--neon-glow);
position: relative;
padding: 10px 20px;
margin-left: 10px;
animation: textGlow 3s infinite alternate;
}
.header-logo::before {
content: '';
position: absolute;
inset: 0;
border: 2px solid transparent;
border-image: linear-gradient(45deg, var(--primary), var(--accent)) 1;
filter: drop-shadow(var(--neon-glow));
animation: borderPulse 4s infinite;
}
.nav-menu {
display: flex;
gap: 20px;
flex-wrap: wrap;
padding-left: 20%;
}
.nav-menu a {
color: white;
text-decoration: none;
font-size: 1.2rem;
font-weight: 500;
padding: 12px 25px;
position: relative;
z-index: 1;
border-radius: 50px;
overflow: hidden;
transition: all 0.4s ease;

```

```

    background: rgba(255, 255, 255, 0.05);
    backdrop-filter: blur(10px);
    border: 1px solid rgba(255, 255, 255, 0.1);
    transform-style: preserve-3d;
    will-change: transform;
}
.nav-menu a:hover {
    color: var(--light);
    transform: rotate(14deg) scale(1.05);
    box-shadow: 0 10px 20px rgba(0, 0, 0, 0.3), var(--neon-glow);
    background: linear-gradient(45deg, rgba(110, 68, 255, 0.3), rgba(255, 68, 168, 0.3));
}
.nav-menu a.active {
    background: linear-gradient(47deg, #7042cd, rgba(229, 70, 155, 0.3));
    animation: pulse 2s infinite;
}
@keyframes float {
    0%, 100% {
        transform: translateY(0) rotateY(0deg);
    }
    50% {
        transform: translateY(-40px) rotateY(10deg);
    }
}
@keyframes textGlow {
    0% {
        text-shadow: 0 0 10px var(--primary), 0 0 20px var(--secondary);
    }
    100% {
        text-shadow: 0 0 20px var(--primary), 0 0 40px var(--secondary), 0 0 60px var(--accent);
    }
}
@keyframes borderPulse {
    0%, 100% {
        opacity: 0.8;
    }
    50% {
        opacity: 0.3;
    }
}
.site-footer {
    position: relative;
    background: linear-gradient(135deg, #f0a20, #1a0933);
    color: #f0eaff;
    padding-top: 80px;
    margin-top: auto;
}
.footer-wave {
    position: absolute;
    top: -1px;
    left: 0;

```

```

width: 100%;
height: 100px;
transform: rotate(180deg);
}
.footer-wave svg {
display: block;
width: 100%;
height: 100%;
}

```

2) Оформление страницы «Главная».

На этой странице представлена аннотация проекта. Она включает в себя цель проекта, ключевые задачи, целевую аудиторию и технологии, применяемые в разработке проекта. Блок аннотации расположен на изображении с виртуальной реальностью, тем самым оно передает саму суть проекта.

Внешний вид главной страницы представлен на рисунке 2.

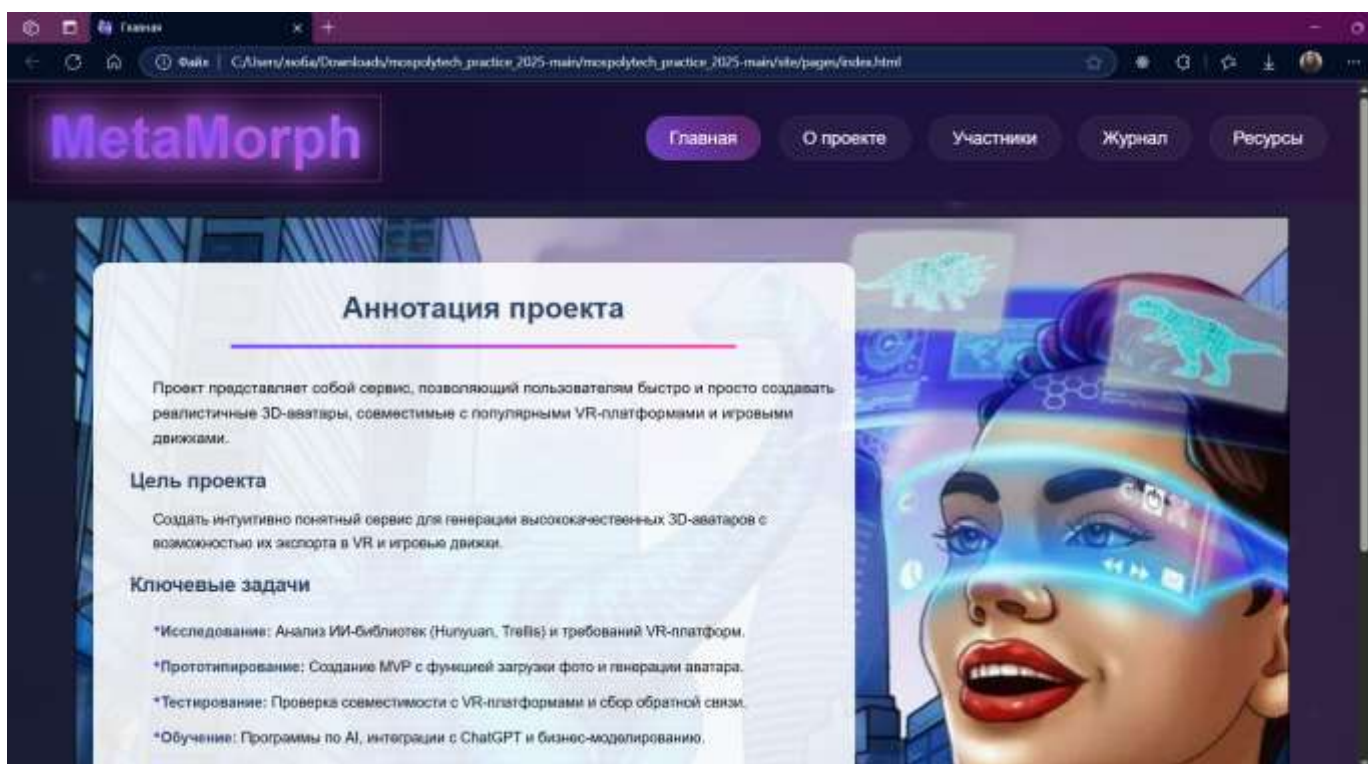


Рисунок 2 – Страница «Главная»

2) Оформление страницы «О проекте».

На этой странице представлена подробная информация о проекте. Здесь было размещено видео с работой приложения, разрабатываемого в рамках

проекта. Также сделаны интерактивные блоки с проблематикой проекта и использованными технологиями. При раскрытии пунктов проблематики проекта можно ознакомиться с более подробным описанием проекта, а при раскрытии технологии можно посмотреть код, который был написан на данной технологии для внедрения в проект, а также прочитать основную информацию о технологии.

Внешний вид страницы представлен на рисунках 3 – 8.

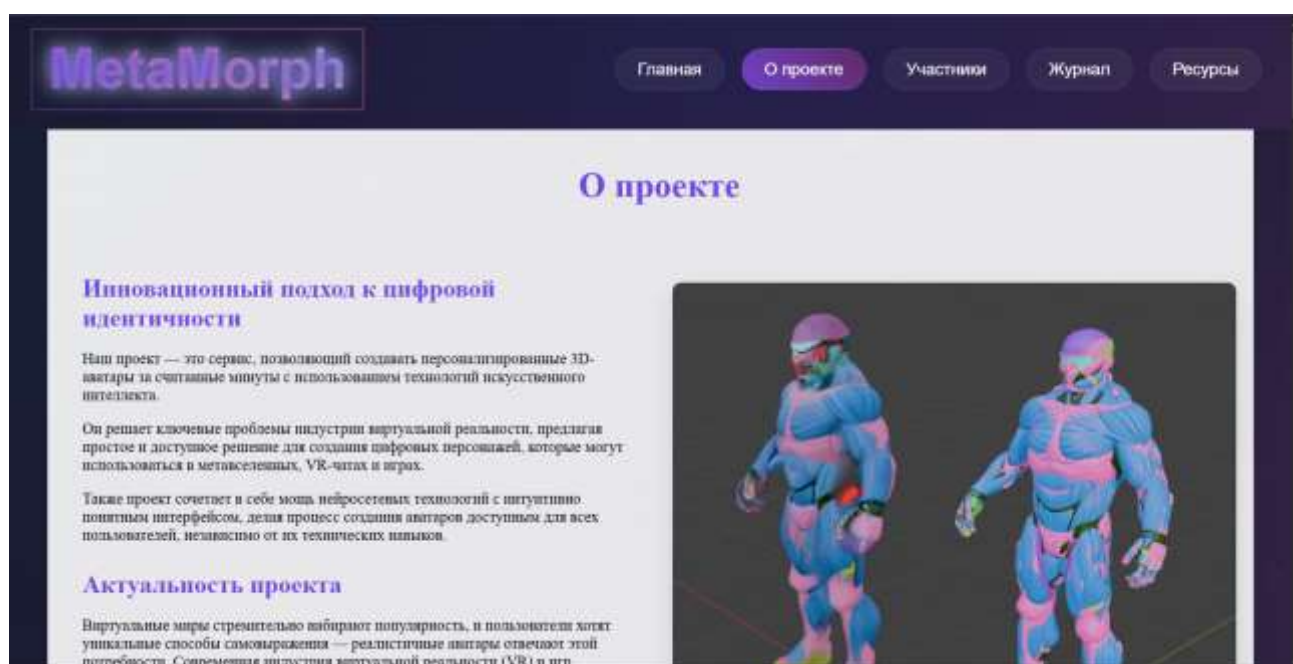


Рисунок 3 – Страница «О проекте»

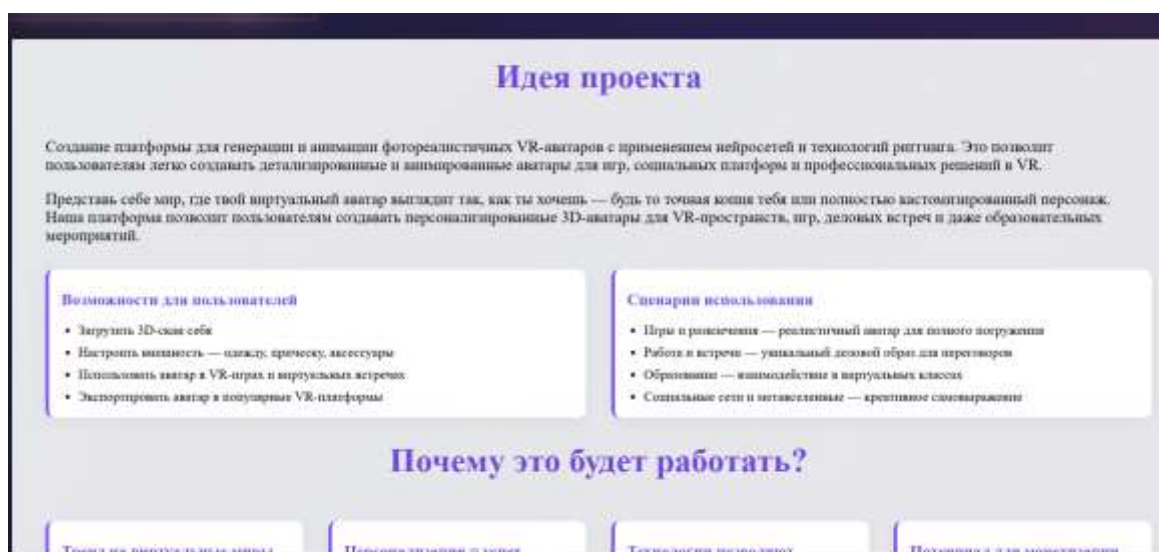


Рисунок 4 – Блок страницы «Идея проекта»

Технологии проекта

Пример кода: Нативный API

```

# Генерация персонажа (фигурка) в формате OBJ
import { Client } from "gradex-client";

const response_0 = await fetch("https://raw.githubusercontent.com/gradex-app/gradex/main/src/example_0.kldbl");
const exampleImage_0 = await response_0.kldbl();
const response_1 = await fetch("https://raw.githubusercontent.com/gradex-app/gradex/main/src/example_1.kldbl");
const exampleImage_1 = await response_1.kldbl();
const response_2 = await fetch("https://raw.githubusercontent.com/gradex-app/gradex/main/src/example_2.kldbl");
const exampleImage_2 = await response_2.kldbl();
const response_3 = await fetch("https://raw.githubusercontent.com/gradex-app/gradex/main/src/example_3.kldbl");
const exampleImage_3 = await response_3.kldbl();
const response_4 = await fetch("https://raw.githubusercontent.com/gradex-app/gradex/main/src/example_4.kldbl");
const exampleImage_4 = await response_4.kldbl();
const client = new Client({apiKey:"xxxxxx@xxxxxx-1"});
const result = await client.predict("/shape-generation", {
  options: "ballbl",
  image: exampleImage_0,
  w_image_front: exampleImage_1,
  w_image_back: exampleImage_2,
  w_image_left: exampleImage_3,
  w_image_right: exampleImage_4,
});
steps: 1,
guidance_scale: 1,

```

Нативный API

Можно считать модель от TensorFlow для обработки ассинхронного языка и генерации 3D-моделей объектов.

Основные функции:

- Генерация ассинхронизированных объектов
- Поддержка русского и английского языков
- Интеграция с экосистемой TensorFlow

Изображения: 3D-модель лица, коллекция 3D-моделей объектов.

© Nude.js

```

graph TD
    A[Прототипирование  
8 нед. - 15 шаг] --> B[Исследование  
12 нед. - 4 шаг]
  
```

Этапы реализации

Проект реализуется в 12 недель по 24 час и включает четыре ключевых этапа

Прототипирование
8 нед. - 15 шаг

- Разработка первой версии интерфейса
- Тестирование моделей и анимаций
- Сбор обратной связи от первых пользователей

Исследование
12 нед. - 4 шаг

- Анализ существующих решений и технологий
- Получение требований VR-платформы и виртуальное движение
- Определение базовой структуры и ключевых потребностей

Рисунок 7 – «Этапы реализации проекта»

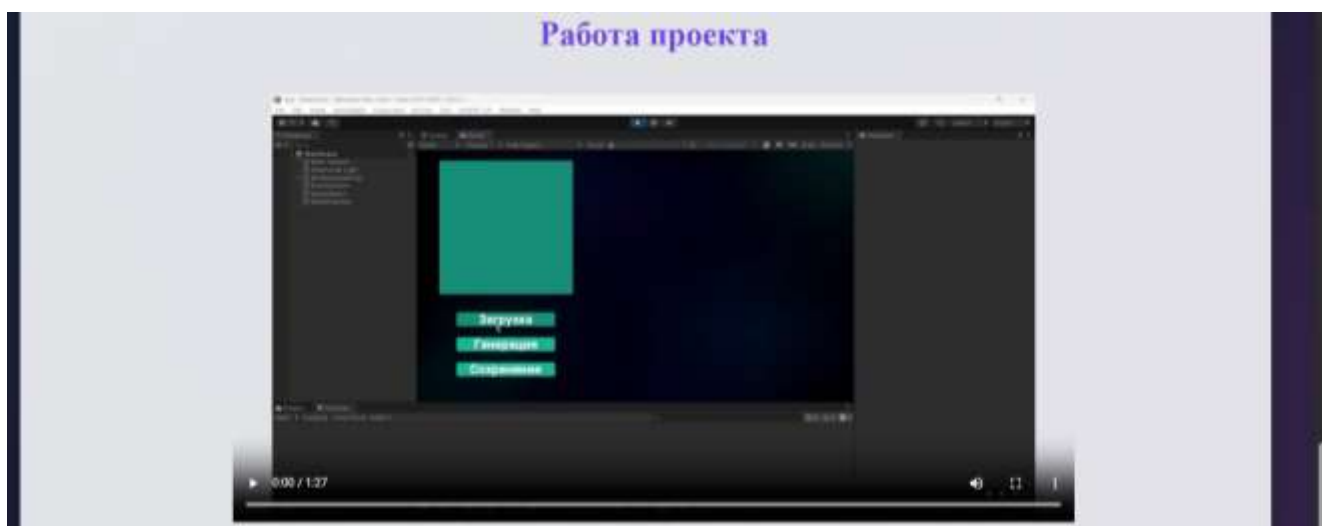


Рисунок 8 – Видео с работой приложения

3) Оформление страницы «Участники».

На этой странице представлены люди, которые совместно разрабатывали проект. Информация о каждом участнике представлена в виде визитной карточки. Переключение между карточками сделано в виде слайдера. Внешний вид страницы представлен на рисунке 9.

Листинг 2 – Слайдер

```
document.addEventListener('DOMContentLoaded', function() {
  const slider = document.getElementById('teamSlider');
  const prevBtn = document.getElementById('prevBtn');
  const nextBtn = document.getElementById('nextBtn');
  const cardWidth = document.querySelector('.member-card').offsetWidth + 30;
  nextBtn.addEventListener('click', function() {
    slider.scrollBy({ left: cardWidth, behavior: 'smooth' });});
  prevBtn.addEventListener('click', function() {
    slider.scrollBy({ left: -cardWidth, behavior: 'smooth' });});
  let isDown = false;
  let startX;
  let scrollLeft;
  slider.addEventListener('mousedown', (e) => {
    isDown = true;
    startX = e.pageX - slider.offsetLeft;
    scrollLeft = slider.scrollLeft;});
  slider.addEventListener('mouseleave', () => {
    isDown = false;});
  slider.addEventListener('mouseup', () => {
    isDown = false;});
  slider.addEventListener('mousemove', (e) => {
```

```

if(!isDown) return;
e.preventDefault();
const x = e.pageX - slider.offsetLeft;
const walk = (x - startX) * 2;
slider.scrollLeft = scrollLeft - walk;
});
});

```

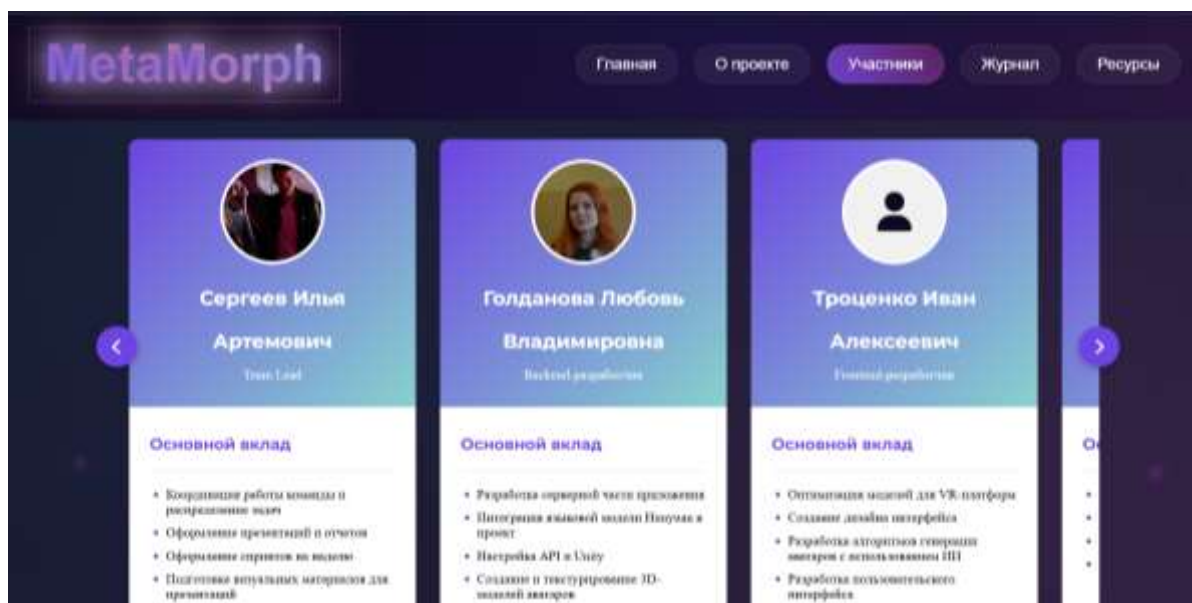


Рисунок 9 – Страница «Участники»

4) Оформление страницы «Журнал».

На этой странице представлен процесс работы над проектом. Внешний вид страницы представлен на рисунке 10.

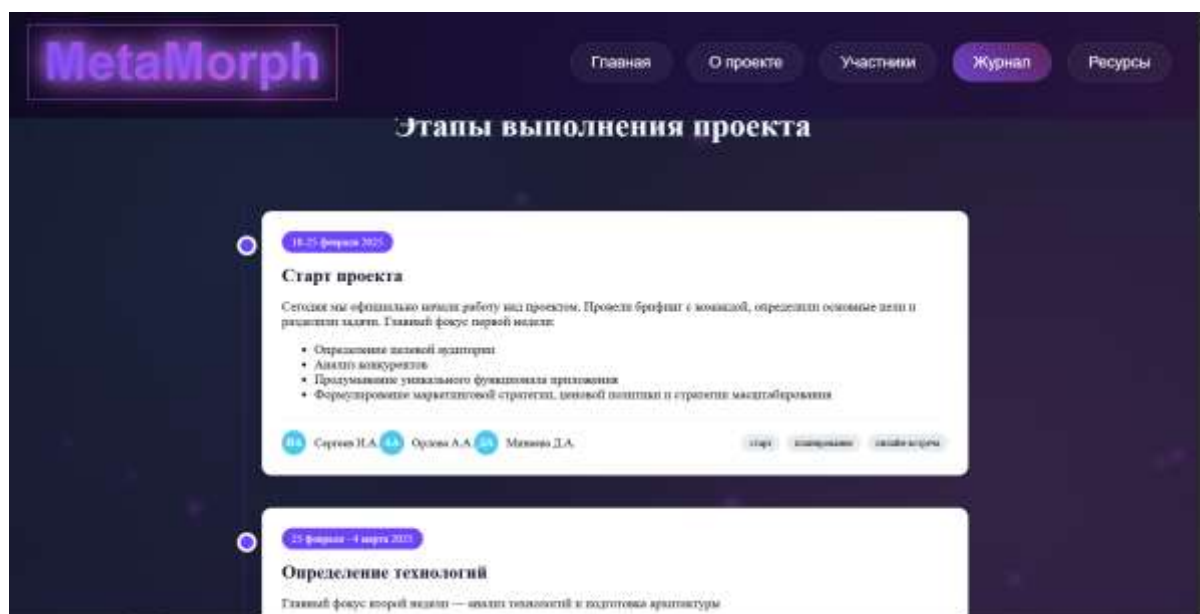


Рисунок 10 – Страница «Журнал»

4) Оформление страницы «Ресурсы».

На этой странице представлены ссылки и ресурсы, которые помогут лучше понять тематику и функционал проекта. Внешний вид страницы представлен на рисунке 11.

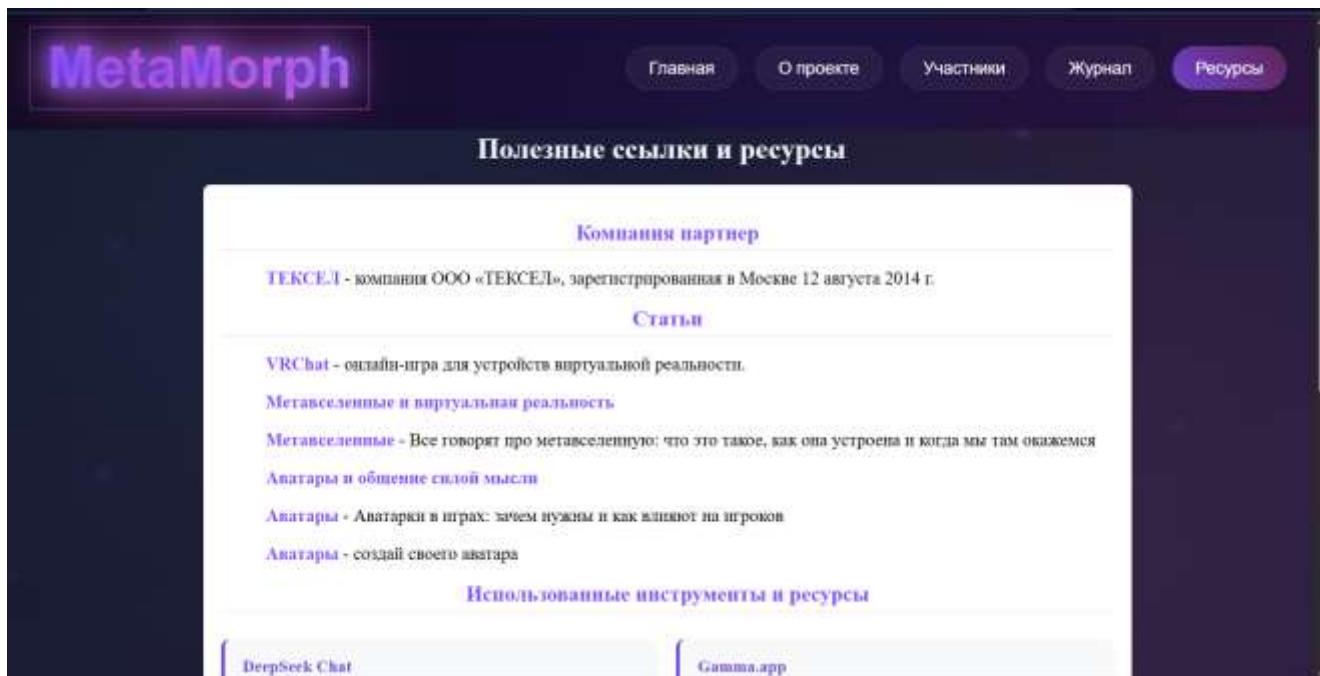


Рисунок 11 – Страница «Ресурсы»

4.2 Результат выполнения вариативной части

В ходе выполнения задания был разработан разработан блокировщик рекламы для браузера на языке JavaScript.

4.2.1 Использованные технологии

JavaScript подходит для написания блокировщика рекламы, потому что позволяет создавать интерактивные расширения для браузеров.

Преимущества JavaScript

- Работа с элементами страницы. JavaScript позволяет находить и удалять рекламные элементы, например, устанавливать свойство `display: none`

- Запускание скриптов. В расширениях для браузера JavaScript-скрипты могут запускаться при загрузке новой страницы или других событиях, что удобно для реализации логики блокировки.
- Кроссбраузерность. Благодаря WebExtension API, JavaScript-код может работать в разных браузерах с минимальными изменениями.

4.2.2 Разработка блокировщика

При создании блокировщика были созданы следующие файлы:

- Manifest.json – необходимый файл, который нужен расширению. Он содержит метаданные о расширении, разрешения, необходимые для работы, и скрипт, который он должен запустить в фоновом режиме.
- Background.js - определяет скрипты, которые должны запускаться при срабатывании определенных событий. Ключевое событие, которое большинство расширений прослушивают в той или иной степени, — это загрузка новой веб-страницы.
- LinkedIn.js – скрипт, в котором прописываются функции блокировки.
- scriptHtml.js – скрипт для связи функционала блокировщика и его внешнего вида, прописанного на html.

После написания основного функционала блокировщик был модифицирован. Ему была выбрана иконка, которая будет отображаться в строке расширений в браузере, а также усовершенствован внешний вид блокировщика. Он представлен на рисунке 12.

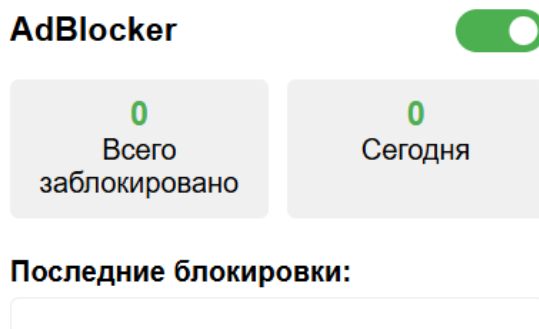


Рисунок 12 – Внешний вид блокировщика

Листинг 3 – HTML код блокировщика

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AdBlocker</title>
  <style>
    body {
      width: 300px;
      padding: 15px;
      font-family: Arial, sans-serif;
      margin: 0;
    }
    .status {
      display: flex;
      align-items: center;
      margin-bottom: 15px;
      justify-content: space-between;
    }
    .switch {
      position: relative;
      display: inline-block;
      width: 50px;
      height: 24px;
    }
    .switch input {
      opacity: 0;
      width: 0;
      height: 0;
    }
    .slider {
      position: absolute;
      cursor: pointer;
      top: 0;
      left: 0;
      right: 0;
      bottom: 0;
      background-color: #ccc;
      transition: .4s;
      border-radius: 24px;
    }
    .slider:before {
      position: absolute;
      content: "";
      height: 16px;
      width: 16px;
      left: 4px;
      bottom: 4px;
      background-color: white;
```

```

    transition: .4s;
    border-radius: 50%;
  }
  input:checked + .slider {
    background-color: #4CAF50;
  }
  input:checked + .slider:before {
    transform: translateX(26px);
  }
  .stats {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 10px;
    margin-bottom: 15px;
  }
  .stat {
    background: #f0f0f0;
    padding: 10px;
    border-radius: 5px;
    text-align: center;
  }
  .stat-value {
    font-size: 18px;
    font-weight: bold;
    color: #4CAF50;
  }
  .blocked-list {
    max-height: 200px;
    overflow-y: auto;
    border: 1px solid #eee;
    border-radius: 5px;
    padding: 10px;
  }
  .blocked-item {
    padding: 5px 0;
    border-bottom: 1px solid #f0f0f0;
    font-size: 12px;
  }
  .blocked-item:last-child {
    border-bottom: none;
  }
</style>
</head>
<body>
  <div class="status">
    <h3 style="margin: 0;">AdBlocker</h3>
    <label class="switch">
      <input type="checkbox" id="toggle" checked>
      <span class="slider"></span>
    </label>
  </div>

```

```

<div class="stats">
  <div class="stat">
    <div class="stat-value" id="totalBlocked">0</div>
    <div>Всего заблокировано</div>
  </div>
  <div class="stat">
    <div class="stat-value" id="todayBlocked">0</div>
    <div>Сегодня</div>
  </div>
</div>
<h4 style="margin-bottom: 5px;">Последние блокировки:</h4>
<div class="blocked-list" id="blockedItems">
</div>
<script src="scriptHtml.js"></script>
</body>
</html>

```

После этого расширение было добавлено в браузер. Для этого были выполнены следующие шаги:

- Выбор «Настройки» в браузере.
- Выбор раздела «Расширения».
- Включение «Режим разработчика», который обычно находится в правом верхнем углу страницы «Расширения».
- Выбор «Загрузить распакованное».
- Выбор каталога созданного проекта в браузере файлов.

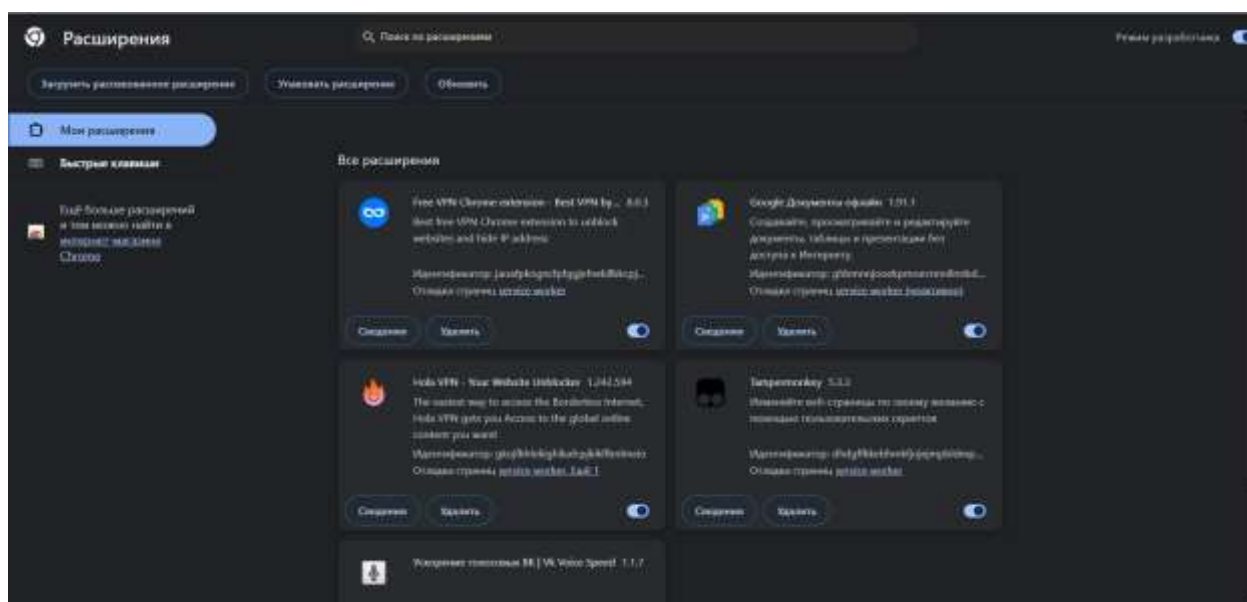


Рисунок 13 – загрузка блокировщика в браузер

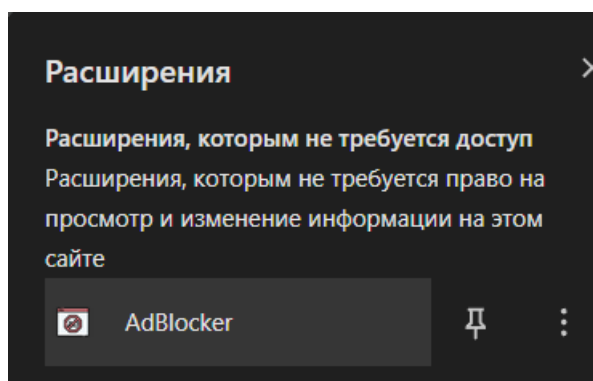


Рисунок 14 – вид с иконкой блокировщика

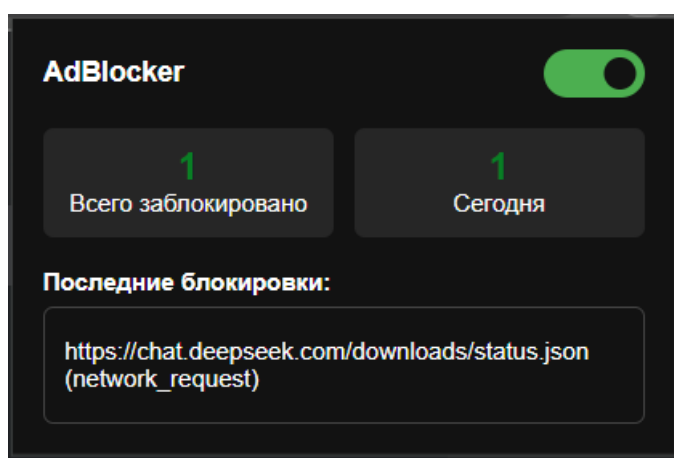


Рисунок 15 – Работа блокировщика

4.3 Ссылки на GitHub

Ознакомиться с кодом web-сайта можно по ссылке:

https://github.com/Lyubov1995/mospolytech_practice_2025/tree/main/site

Ознакомиться с кодом блокировщика можно по ссылке:

https://github.com/Lyubov1995/mospolytech_practice_2025/tree/main/src

ЗАКЛЮЧЕНИЕ

Проект по созданию сервиса для генерации 3D-аватаров, совместимых с VR-платформами и игровыми движками, представляет собой важный шаг в развитии технологий виртуальной реальности и искусственного интеллекта. Реализация данного сервиса поможет пользователям и предпринимателям преодолеть барьеры, связанные с ограниченными возможностями персонализации и высоким порогом входа в сферу VR и 3D-моделирования.

В ходе разработки проекта решены ключевые проблемы, такие как недостаточная выразительность аватаров и сложности с созданием высококачественных 3D-моделей. Проект ориентирован не только на пользователей, но и на начинающих предпринимателей, предлагая им обучение и поддержку в области VR, AI и стартапов, что создаст дополнительные возможности для карьерного роста и дальнейшего развития технологий.

Результаты включают создание удобного и функционального инструмента для пользователей, обучение участников основам VR-бизнеса и привлечение первых партнеров и инвесторов.

В результате выполнения проектной практики были выполнены следующие задачи:

- 1) Настроен Git и репозиторий.
- 2) Написаны документы в Markdown.
- 3) Создан статический веб-сайта
- 4) Разработан собственный блокировщик рекламы для браузера на JavaScript.

СПИСОК ЛИТЕРАТУРЫ

1. Введение в CSS верстку. [сайт] — URL: https://developer.mozilla.org/ru/docs/Learn_web_development/Core/CSS_layout/Introduction Режим доступа: свободный — Текст: электронный.
2. DevTools для «чайников»: [сайт] — URL: <https://habr.com/ru/articles/548898/> Режим доступа: свободный — Текст: электронный.
3. Элементы HTML: [сайт] — URL: <https://developer.mozilla.org/ru/docs/Web/HTML/Element> Режим доступа: свободный — Текст: электронный
4. Основы HTML: [сайт] — URL: [https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your first website/Creating the content](https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content) Режим доступа: свободный — Текст: электронный
5. Основы CSS: [сайт] — URL: <https://developer.mozilla.org/ru/docs/Web/CSS> Режим доступа: свободный — Текст: электронный
6. Официальная документация Git: [сайт] — URL: <https://git-scm.com/book/ru/v2> Режим доступа: свободный — Текст: электронный
7. Уроки по Markdown: [сайт] — URL: https://ru.hexlet.io/lesson_filters/markdown Режим доступа: свободный — Текст: электронный
8. Руководство по JavaScript [сайт] — URL: [Руководство по JavaScript](#) Режим доступа: свободный — Текст: электронный