

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
«Название проекта»

по дисциплине «Проектный практикум»

Заказчик: Фамилия И.О.

Шестеров Михаил

Андреевич

Куратор: Фамилия И.О.

Шестеров Михаил

ученая степень, ученое звание, должность

Андреевич

Студенты команды: AeroGuard

Фамилия И.О.

Мостафа Мохамед

Ашраф Якут Саад

Фамилия И.О.

Гаттс Абануб Хамис

Даниал

Фамилия И.О.

Отман Али Амр Али

Мохамед

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

.....	3
1 Основная часть	4
1.1 Анализ требований заказчика и планирование проекта	
.....	4
1.2 Анализ аналогов и конкурентов	
.....	5
1.3 Архитектура программного продукта	
.....	6
1.3.1 Общая архитектура системы	
6	
1.3.2 Фронтенд-архитектура	
7	
1.3.3 Структура проекта	
8	
1.4 Методология разработки и процесс разработки	
.....	9
1.5 Реализация ключевых компонентов	
.....	10
1.5.1 Модуль симуляции космического мусора	
10	
1.5.2 Компонент панели мониторинга	
11	
1.5.3 Система навигации и маршрутизации	
12	
1.6 Пользовательский интерфейс и взаимодействие	
.....	13
1.7 Тестирование и обеспечение качества	
.....	14
1.8 Распределение задач между участниками команды	
.....	15

ЗАКЛЮЧЕНИЕ	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	17
ПРИЛОЖЕНИЕ А (обязательное) Структура файлов проекта	18

ВВЕДЕНИЕ

Проблема засорения околоземного космического пространства является одной из наиболее актуальных проблем современной космонавтики. Количество объектов космического мусора постоянно увеличивается, что создает серьезную угрозу для функционирующих спутников, космических станций и будущих космических миссий. Согласно данным различных космических агентств, на орбите Земли находится более 34 000 отслеживаемых объектов размером свыше 10 см и миллионы более мелких фрагментов.

Целью данного проекта является разработка веб-приложения для симуляции и мониторинга космического мусора, которое обеспечивает:

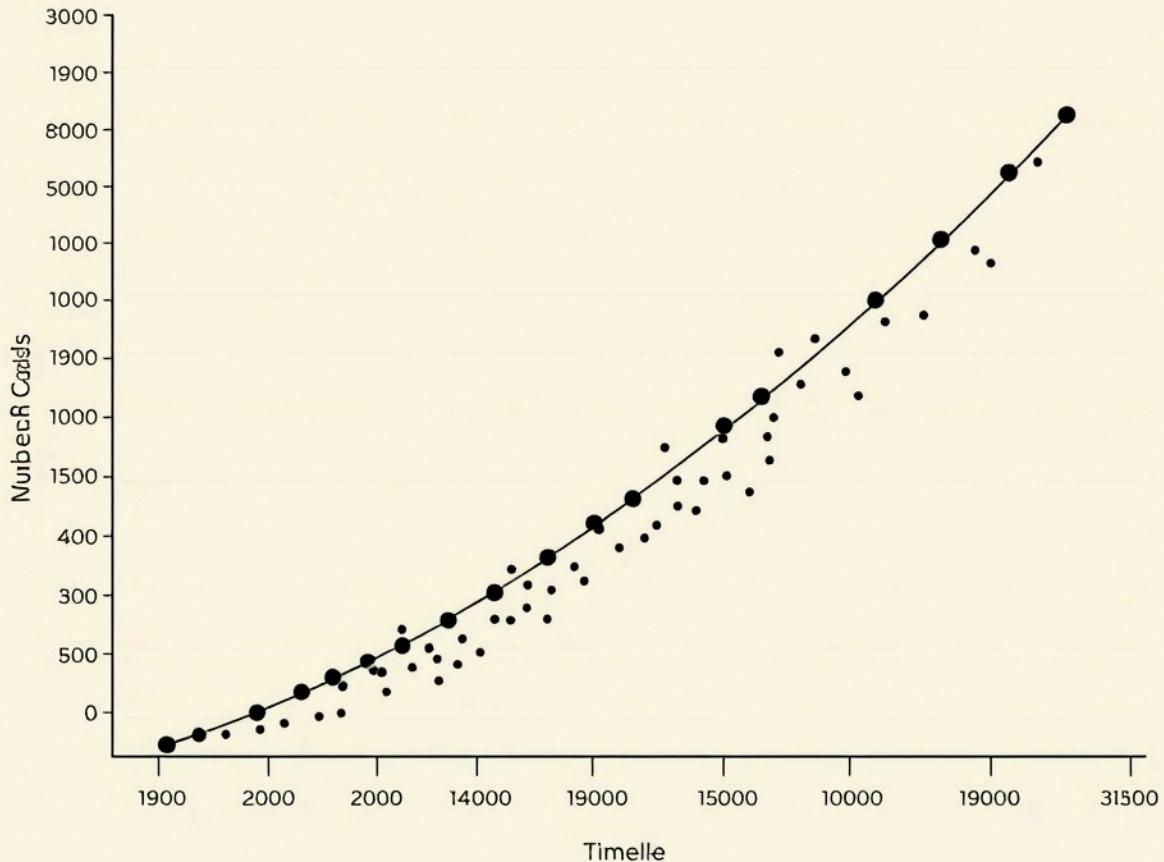
- а) визуализацию движения объектов космического мусора в режиме реального времени;
- б) мониторинг ключевых характеристик и параметров космических объектов;
- в) интуитивно понятный пользовательский интерфейс для анализа данных;
- г) масштабируемую архитектуру для дальнейшего развития функциональности.

Актуальность проекта обусловлена возрастающей потребностью в инструментах для анализа и прогнозирования поведения космического мусора. Разработанное приложение может применяться в образовательных целях для демонстрации проблемы космического

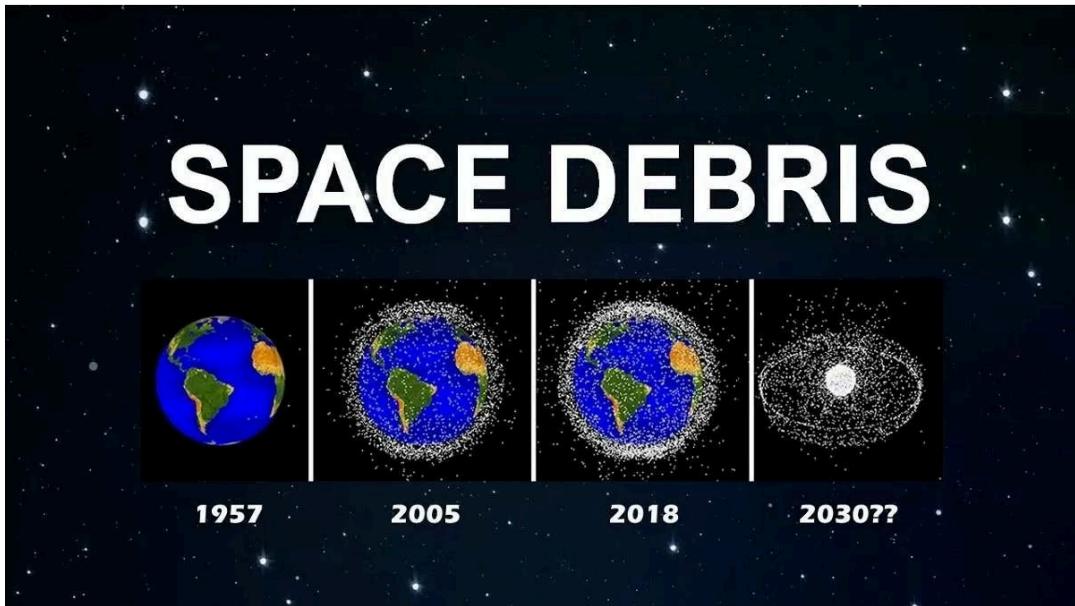
мусора, в исследовательских целях для анализа распределения объектов на орбите, а также как основа для более сложных систем мониторинга космического пространства.

Ожидаемыми результатами проекта являются создание функционального веб-приложения с возможностью интерактивной визуализации космических объектов, реализация системы мониторинга с отображением статистических данных, обеспечение адаптивного дизайна для различных устройств и создание масштабируемой архитектуры для дальнейшего развития.

Growth of Celestlallosss över



■ The'reasment Fasign.com



1 Основная часть

1.1 Анализ требований заказчика и планирование проекта

Анализ требований заказчика выявил необходимость создания веб-приложения, которое должно решать следующие основные задачи: обеспечение визуализации движения космического мусора, предоставление инструментов мониторинга и анализа данных, создание интуитивного пользовательского интерфейса.

На основе анализа требований был составлен backlog проекта, включающий следующие основные пользовательские истории:

- а) как пользователь, хочу видеть трехмерную визуализацию космических объектов для понимания их расположения в пространстве;
- б) как пользователь, хочу получать актуальную информацию о характеристиках космического мусора для анализа текущей ситуации;
- в) как пользователь, хочу использовать приложение на различных устройствах для обеспечения мобильности;
- г) как пользователь, хочу интуитивно понятную навигацию для эффективного взаимодействия с системой.

План действий для достижения цели включал этапы проектирования архитектуры, разработки основных компонентов, интеграции модулей, тестирования и оптимизации производительности.

Планировщик проекта AeroGuardV3						
		Due Date	Assignee	Com...	Comp...	Compl...
+ Add Task	Hide					
<input checked="" type="checkbox"/> Индикатор радиуса визуального радара					5/1/2025	moheda438
<input checked="" type="checkbox"/> Интерфейс команд и управления					5/1/2025	moheda438
<input checked="" type="checkbox"/> Управление скоростью симуляции					5/1/2025	moheda438
<input checked="" type="checkbox"/> Регулировка дальности радара					5/1/2025	moheda438
<input checked="" type="checkbox"/> Ручные элементы управления навигацией					5/1/2025	moheda438
<input checked="" type="checkbox"/> Переключение между ручными и режимами с поддержкой ИИ					5/1/2025	moheda438
<input checked="" type="checkbox"/> Система навигации ИИ					5/1/2025	moheda438
<input checked="" type="checkbox"/> Алгоритм избегания мусора					5/1/2025	moheda438
<input checked="" type="checkbox"/> Функция отслеживания целей					5/1/2025	moheda438
<input checked="" type="checkbox"/> Предупреждение о предстоящем столкновении					5/1/2025	moheda438
<input checked="" type="checkbox"/> Телеметрия и визуализация данных					5/1/2025	moheda438
<input checked="" type="checkbox"/> Метрики космического корабля в реальном времени					5/1/2025	moheda438
<input checked="" type="checkbox"/> Панель предупреждений об опасности					5/1/2025	moheda438
<input checked="" type="checkbox"/> Панель информации об объектах					5/1/2025	moheda438
<input checked="" type="checkbox"/> Оценка риска столкновения					5/1/2025	moheda438
+ Add Task						
Техническая реализация						
<input checked="" type="checkbox"/> Настройка основной структуры					5/1/2025	moheda438

1.2 Анализ аналогов и конкурентов

Анализ существующих решений в области визуализации космических объектов показал наличие нескольких категорий продуктов.

Профессиональные системы мониторинга космического пространства, используемые космическими агентствами, обладают высокой точностью, но имеют сложный интерфейс и высокую стоимость.

Образовательные приложения и симуляторы отличаются простотой использования, но ограниченным функционалом. Веб-сервисы для

отслеживания спутников предоставляют доступность через браузер, но фокусируются на активных спутниках, а не на мусоре.

Сравнительный анализ показал, что существующие решения либо слишком сложны для широкого использования, либо не покрывают специфические требования к визуализации космического мусора. Это подтверждает актуальность разработки специализированного веб-приложения, сочетающего доступность, функциональность и визуальную привлекательность.

Конкурентными преимуществами разрабатываемого решения являются специализация на космическом мусоре, современный веб-интерфейс, адаптивный дизайн и модульная архитектура.

1.3 Архитектура программного продукта

1.3.1 Общая архитектура системы

Архитектура приложения построена по принципу одностраничного веб-приложения (SPA) с клиент-серверной моделью. Клиентская часть реализована как SPA с использованием современных веб-технологий, обеспечивающих интерактивность и отзывчивость интерфейса.

Серверная часть предназначена для предоставления данных через API, хотя в текущей версии акцент сделан на клиентской реализации.

Выбор архитектурного решения обусловлен требованиями к интерактивности пользовательского интерфейса, необходимостью обеспечения быстрого отклика системы на действия пользователя, возможностью масштабирования функциональности без изменения основной архитектуры.

1.3.2 Фронтенд-архитектура

Фронтенд приложения построен на основе библиотеки React с использованием TypeScript для обеспечения типобезопасности.

Архитектура следует компонентному подходу, где каждый элемент интерфейса представлен отдельным компонентом, что обеспечивает модульность и возможность повторного использования кода.

Основными принципами архитектуры являются разделение ответственности между компонентами, использование React Hooks для управления состоянием, применение современных паттернов разработки. Выбор React обусловлен его зрелостью, обширной экосистемой, хорошей производительностью и широкой поддержкой сообщества. TypeScript добавляет статическую типизацию, что значительно упрощает разработку и снижает количество ошибок во время выполнения.

1.3.3 Структура проекта

Структура проекта организована в соответствии с принципами модульности и логического разделения компонентов:

- а) каталог src/ содержит весь исходный код приложения;
- б) src/components/ включает переиспользуемые компоненты интерфейса;
- в) src/pages/ содержит компоненты страниц приложения;
- г) src/hooks/ включает пользовательские React-хуки;
- д) src/lib/ содержит вспомогательные утилиты и библиотеки;
- е) src/components/ui/ включает базовые UI-компоненты библиотеки Shadcn.

Такая организация обеспечивает легкость навигации по проекту, упрощает поиск и модификацию компонентов, способствует соблюдению принципов чистого кода.

```
1 import { Link } from "react-router-dom";
2 import { Shield, Satellite, Activity, AlertTriangle, BarChart4, Rocket } from "lucide-react";
3 import { Card } from "@components/ui/card";
4 import { Button } from "@components/ui/button";
5
6 Qodo Gen: Options | Test this function
7 const Index = () => {
8   return (
9     <div className="flex-1 starfield">
10       <div className="container mx-auto px-4 py-8">
11         <div className="max-w-4xl mx-auto text-center mb-12">
12           <h1 className="text-4xl font-bold mb-4 text-white">
13             Space Defense Simulation System
14           </h1>
15           <p className="text-lg text-gray-300 mb-6">
16             Monitor, detect, and respond to space debris threats with our
17             advanced simulation platform
18           </p>
19           <Link to="/simulation">
20             <Button className="bg-space-accent hover:bg-space-accent/80 text-black font-medium">
21               Launch Simulation
22             </Button>
23           </Link>
24         </div>
25
26         <div className="grid md:grid-cols-3 gap-6 mb-12">
27           <Card className="bg-space-panel border-space-grid p-6">
28             <div className="flex justify-between items-start mb-4">
29               <Satellite className="text-space-accent h-8 w-8" />
30               <span className="text-sm text-space-accent bg-space-accent/10 px-2 py-1 rounded">Real-time</span>
31             </div>
32             <h3 className="text-xl font-medium mb-2">Live Satellite Tracking</h3>
33             <p className="text-gray-400 text-sm">
34               Monitor satellite positions using real-time TLE data with accurate orbital predictions
35             </p>
36           </Card>
37
38           <Card className="bg-space-panel border-space-grid p-6">
39             <div className="flex justify-between items-start mb-4">
40               <AlertTriangle className="text-space-warning h-8 w-8" />
41               <span className="text-sm text-space-warning bg-space-warning/10 px-2 py-1 rounded">Predictive</span>
42             </div>
43             <h3 className="text-xl font-medium mb-2">Collision Prediction</h3>
44             <p className="text-gray-400 text-sm">
45               Advanced algorithms to predict potential collisions and provide timely alerts
46             </p>
47           </Card>
48
49         </div>
50       </div>
51     </div>
52   )
53 
```

1.4 Методология разработки и процесс разработки

В проекте применялась итеративная методология разработки с элементами Agile. Процесс разработки был разделен на несколько итераций, каждая из которых включала планирование, реализацию, тестирование и анализ результатов.

Первая итерация была посвящена созданию базовой структуры приложения и настройке инфраструктуры разработки. На этом этапе были настроены инструменты сборки проекта с использованием Vite, конфигурация TypeScript, подключение библиотек компонентов Shadcn UI, создание базовой структуры маршрутизации.

Вторая итерация включала разработку основных компонентов визуализации. Основное внимание уделялось созданию компонента симуляции, реализации базовой логики отображения объектов, интеграции с системой навигации.

Третья итерация была сосредоточена на создании панели мониторинга и улучшении пользовательского интерфейса. Разрабатывался компонент

DebrisDashboard, реализовывалось отображение статистических данных, проводилась оптимизация дизайна для различных устройств.

Четвертая итерация включала тестирование, оптимизацию и подготовку к развертыванию. Проводились комплексные тесты функциональности, оптимизация производительности, подготовка документации.

1.5 Реализация ключевых компонентов

1.5.1 Модуль симуляции космического мусора

Модуль симуляции представлен компонентом *SimulationCanvas*, который отвечает за визуализацию движения космических объектов. Компонент реализует интерактивную трехмерную сцену, позволяющую пользователю наблюдать за движением объектов космического мусора в режиме реального времени.

Ключевые особенности реализации включают использование современных веб-технологий для создания трехмерной графики, реализацию физически корректных моделей движения объектов, обеспечение интерактивности через элементы управления камерой и масштабированием.

Модуль поддерживает различные режимы отображения, включая статический режим для детального анализа положения объектов и динамический режим для наблюдения за движением в реальном времени. Реализованы элементы управления для изменения параметров визуализации, что позволяет пользователю настраивать отображение в соответствии со своими потребностями.

1.5.2 Компонент панели мониторинга

Компонент DebrisDashboard представляет собой информационную панель, отображающую ключевые характеристики и статистические данные о космическом мусоре. Панель обеспечивает централизованный доступ к важной информации и позволяет пользователю быстро оценивать текущую ситуацию.

Функциональность панели включает отображение общей статистики по количеству объектов различных типов, визуализацию распределения

объектов по орбитальным характеристикам, представление информации о потенциально опасных объектах, отображение временных трендов изменения количества мусора.

Интерфейс панели построен с использованием компонентов Shadcn UI, что обеспечивает современный и профессиональный внешний вид.

Реализована адаптивная верстка, позволяющая панели корректно отображаться на устройствах различных размеров.

1.5.3 Система навигации и маршрутизации

Система навигации реализована с использованием React Router и обеспечивает плавное переключение между различными разделами приложения. Основными страницами являются главная страница с обзором функциональности, страница симуляции с интерактивной визуализацией, страница ошибки 404 для обработки некорректных маршрутов.

Компонент Header содержит основные элементы навигации и обеспечивает единообразный интерфейс для перемещения по приложению. Навигация спроектирована с учетом принципов удобства использования и интуитивности.

Controls
Dashboard

DEBRIS ANALYTICS DASHBOARD

Current Debris Metrics

Total Objects: 0 Nearby Objects: 0

Risk Level: Low Last Updated: 8:56:29 PM

Debris Properties

Average Velocity: 2.11 km/s Average Weight: 211.8 kg Average Size: 1.0 m

Debris Count History

Time	Total Debris
20:26	22
20:27	20
20:28	25
20:29	28
20:30	20
20:31	22
20:32	25
20:33	20
20:34	28
20:35	22
20:36	25
20:37	20
20:38	28
20:39	22
20:40	25
20:41	20
20:42	28
20:43	22
20:44	25
20:45	20
20:46	28
20:47	22
20:48	25

Velocity Trends

Time	Velocity (km)
20:26	3.5
20:27	2.5
20:28	4.0
20:29	3.0
20:30	5.5
20:31	2.0
20:32	5.0
20:33	3.5
20:34	4.5
20:35	2.5
20:36	5.5
20:37	1.5
20:38	4.0
20:39	3.0
20:40	5.0
20:41	2.0
20:42	4.5
20:43	3.0
20:44	5.5
20:45	2.0
20:46	4.0
20:47	3.0
20:48	5.0

Size & Weight Distribution

Time	Weight (kg)	Size (m)
20:47	200	0.5
20:49	230	0.4
20:51	210	0.6
20:53	180	0.3
20:55	200	0.5

TELEMETRY

Spacecraft

Velocity: 0 km/s Altitude: 400 km Fuel: 98% Shields: 100%

Environment

Radar Range: 15 km Debris Count: 0 Nearby Objects: 0 Solar Activity: High

Safety

Collision Risk: Low Radiation Level: Elevate

System

AI Status: Inactive System Integrity: 98%

COMMAND CORE

Simulation Speed: ↻ Radar Range: 15 km Simulation Status: Running (1x)

NAVIGATION HUB

↓ Blow Down
←
↑
→

Use controls to navigate spacecraft

AI NEXUS

AI Control
Manual

⟲ Off
○ Avoid
◎ Follow

● AI navigation disabled

1.6 Пользовательский интерфейс и взаимодействие

Пользовательский интерфейс приложения разработан с акцентом на удобство использования и информативность. Дизайн следует современным принципам веб-дизайна и обеспечивает интуитивное взаимодействие пользователя с системой.

Основные принципы дизайна включают минимализм и четкость в представлении информации, использование визуальной иерархии для выделения важных элементов, обеспечение быстрого доступа к ключевым функциям, адаптивность для различных размеров экранов.

Цветовая схема и типографика выбраны с учетом специфики космической тематики и обеспечивают хорошую читаемость информации. Использование библиотеки Shadcn UI гарантирует единообразность визуальных элементов и соответствие современным стандартам веб-дизайна.

Интерактивные элементы снабжены визуальной обратной связью, что улучшает пользовательский опыт. Реализованы подсказки и дополнительная информация для сложных элементов интерфейса.

Space Defense Simulation System

Monitor, detect, and respond to space debris threats with our advanced simulation platform

[Launch Simulation](#)

Real-time

Live Satellite Tracking

Monitor satellite positions using real-time TLE data with accurate orbital predictions



Predictive

Collision Prediction

Advanced algorithms to predict potential collisions and provide timely alerts



Data Analytics

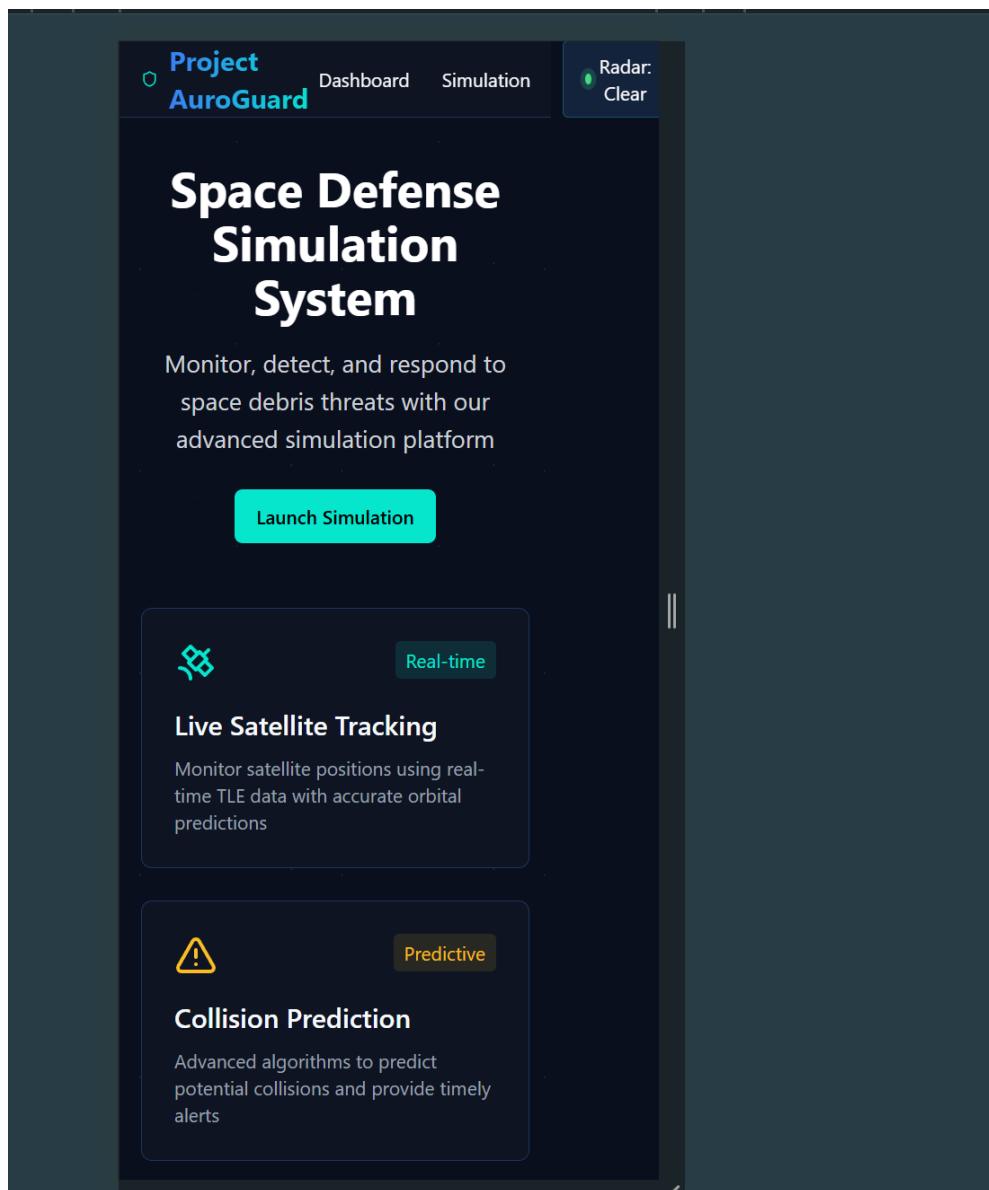
Historical logs and analytics to inform collision avoidance strategies

Interactive Simulation Environment

Experience a realistic simulation of how satellites detect and avoid space debris using our advanced visualization platform.

- ⌚ Real-time debris tracking
- ⚡ Advanced collision avoidance systems
- 🚀 Spacecraft maneuver simulation

[Enter Simulation](#)



1.7 Тестирование и обеспечение качества

Процесс тестирования включал несколько уровней проверки качества разработанного программного продукта. Модульное тестирование проводилось для проверки корректности работы отдельных компонентов и функций, что позволило выявить и устранить ошибки на раннем этапе разработки.

Интеграционное тестирование включало проверку взаимодействия между различными модулями приложения, тестирование API для

обмена данными, проверку корректности навигации между страницами, тестирование совместимости компонентов.

Функциональное тестирование проводилось для проверки соответствия реализованной функциональности заявленным требованиям.

Тестились основные пользовательские сценарии, проверялась корректность отображения данных, анализировалась производительность при различных нагрузках.

Тестирование пользовательского интерфейса включало проверку корректного отображения элементов на различных устройствах и в различных браузерах, тестирование адаптивности дизайна, проверку удобства взаимодействия с интерфейсом.

В результате тестирования были выявлены и устраниены критические ошибки, оптимизирована производительность приложения, улучшена стабильность работы системы.

1.8 Распределение задач между участниками команды

Планирование деятельности в ходе разработки включало распределение ролей и ответственности между участниками команды. Каждый участник получил четко определенные зоны ответственности, что обеспечило эффективную параллельную работу над различными компонентами системы.

Первый участник команды отвечал за разработку архитектуры приложения и настройку инфраструктуры разработки, создание базовых компонентов и системы маршрутизации, координацию работы команды и интеграцию модулей.

Второй участник команды занимался разработкой модуля симуляции и визуализации, реализацией интерактивных элементов трехмерной графики, оптимизацией производительности графических компонентов.

Третий участник команды был ответственен за создание пользовательского интерфейса и компонента панели мониторинга, реализацию адаптивного дизайна, проведение тестирования удобства использования.

Координация работы осуществлялась через регулярные встречи команды для обсуждения прогресса и планирования следующих шагов, использование системы контроля версий для совместной работы с кодом, документирование принятых решений и изменений в архитектуре.

ЗАКЛЮЧЕНИЕ

Разработанное веб-приложение для симуляции и мониторинга космического мусора полностью соответствует поставленным требованиям заказчика. Реализованная функциональность обеспечивает интерактивную визуализацию движения космических объектов, предоставляет инструменты для мониторинга ключевых характеристик мусора, включает интуитивно понятный пользовательский интерфейс.

Анализ выполненных требований показывает успешную реализацию всех основных пользовательских историй. Трехмерная визуализация космических объектов обеспечивает наглядное представление проблемы космического мусора. Панель мониторинга предоставляет структурированную информацию о характеристиках объектов. Адаптивный дизайн позволяет использовать приложение на различных устройствах.

Оценка качества программного продукта на основе результатов тестирования выявила высокую стабильность работы системы. Модульное и интеграционное тестирование не выявило критических ошибок, влияющих на работоспособность продукта. Функциональное тестирование подтвердило корректность реализации основных сценариев использования. Тестирование пользовательского интерфейса показало хорошую адаптивность и удобство взаимодействия.

Предложения по улучшению продукта включают интеграцию с реальными базами данных космического мусора для получения актуальной информации, реализацию более сложных физических моделей симуляции для повышения точности прогнозирования,

добавление функций анализа рисков столкновений, расширение возможностей панели мониторинга для детального анализа данных.

Предпосылки для развития продукта включают возможность масштабирования архитектуры для поддержки большего количества пользователей, потенциал интеграции с профессиональными системами мониторинга космического пространства, возможность использования в образовательных программах по космическим технологиям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. React – A JavaScript library for building user interfaces [Электронный ресурс] / Facebook Inc. – URL: <https://reactjs.org> (дата обращения: 20.05.2025).
2. TypeScript Documentation [Электронный ресурс] / Microsoft Corporation. – URL: <https://www.typescriptlang.org/docs> (дата обращения: 20.05.2025).
3. Shadcn/ui [Электронный ресурс]. – URL: <https://ui.shadcn.com> (дата обращения: 20.05.2025).
4. Vite [Электронный ресурс]. – URL: <https://vitejs.dev> (дата обращения: 20.05.2025).
5. European Space Agency. Space Debris [Электронный ресурс] / ESA. – URL: https://www.esa.int/Safety_Security/Space_Debris (дата обращения: 20.05.2025).
6. NASA. Orbital Debris Program Office [Электронный ресурс] / NASA. – URL: <https://orbitaldebris.jsc.nasa.gov> (дата обращения: 20.05.2025).

ПРИЛОЖЕНИЕ А (обязательное)

Структура файлов проекта

Структура файлов разработанного веб-приложения организована следующим образом:

project-root/

```
|   └── src/
|       ├── App.tsx
|       ├── main.tsx
|       └── pages/
|           ├── Index.tsx
|           ├── NotFound.tsx
|           └── Simulation.tsx
|       └── components/
|           ├── DebrisDashboard.tsx
|           ├── Header.tsx
|           ├── SimulationCanvas.tsx
|           └── ui/
|               ├── button.tsx
|               ├── card.tsx
|               ├── table.tsx
|               └── [другие UI компоненты]
|       └── hooks/
|           └── use-mobile.ts
|       └── lib/
|           └── utils.ts
└── public/
    └── index.html
```

```
├── package.json  
├── tsconfig.json  
├── vite.config.ts  
└── [конфигурационные файлы]
```

Описание основных каталогов и файлов:

src/App.tsx – основной компонент приложения, содержащий корневую логику и маршрутизацию.

src/main.tsx – точка входа в приложение, инициализация React-приложения.

src/pages/ – каталог компонентов страниц приложения.

src/components/ – каталог переиспользуемых компонентов интерфейса.

src/components/ui/ – каталог базовых UI-компонентов библиотеки Shadcn.

src/hooks/ – каталог пользовательских React-хуков.

src/lib/ – каталог вспомогательных утилит и библиотек.

Project AuroGuard

Space Defense Simulation System

Monitor, detect, and respond to space debris threats with our advanced simulation platform

[Launch Simulation](#)



Live Satellite Tracking

Real-time

Monitor satellite positions using real-time TLE data with accurate orbital predictions



Collision Prediction

Predictive

Advanced algorithms to predict potential collisions and provide timely alerts



Data Analytics

Analytics

Historical logs and analytics to improve future collision avoidance strategies

Interactive Simulation Environment

Experience a realistic simulation of how satellites detect and avoid space debris using our advanced visualization platform.

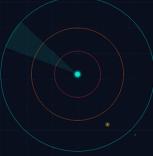
-  Real-time debris tracking
-  Advanced collision avoidance systems
-  Spacecraft maneuver simulation

[Enter Simulation](#)



Project AuroGuard

Zoom: 1.0x

TELEMETRY

Spacecraft	Environment
Velocity2	Radar Range
km/s	15 km
Altitude	Debris Count
400 km	1
Fuel	Nearby Objects
100%	Solar Activity
Shields	
100%	High

Safety

Collision Risk: Low

System

AI Status: Inactive

Radiation Level: Elevate

System Integrity: 99%

COMMAND CORE

Simulation Speed: Fast

Radio Range: 15 km

Simulation Status: Running (1x)

NAVIGATION HUB

Use controls to navigate spacecraft

Controls: Slow Down, Up, Down, Left, Right

AI NEXUS

AI Control: Manual

Solar Activity Warning: Increased solar radiation detected. Shield monitoring advised.

