

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Кафедра информационных систем

КУРСОВАЯ РАБОТА

Тема: Разработка веб-приложения "IMarket"

Работу выполнил студент: Джахан Исрат группы М33091
(фамилия, имя, отчество) (номер группы)

Руководитель Андреев Николай Владимирович, Университет ИТМО,
преподаватель (квалификационная категория "преподаватель практики")
(фамилия, имя, отчество)

Работа защищена " _____ " _____ 2024 г. с оценкой _____

Подписи членов комиссии: _____

САНКТ-ПЕТЕРБУРГ
2024

УНИВЕРСИТЕТ ИТМО

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

Студент	<u>Джахан Исрат</u>		
	(Фамилия, И., О.)		
Факультет	<u>Информационных технологий и программирования</u>		
Кафедра	<u>Информационных систем</u>	Группа	<u>М33091</u>
Направление (специальность)	<u>Информационные системы и технологии</u>		
Руководитель	<u>Андреев Николай Владимирович, Университет ИТМО, преподаватель</u> <u>(квалификационная категория "преподаватель практики")</u> (Фамилия, И.О., должность, ученое звание, степень)		
Дисциплина	<u>Web-программирование</u>		
Наименование темы	<u>«Разработка веб-приложения "IMarket"»</u>		
Задание	<u>Разработать веб-приложение, проанализировать и смоделировать процессы, средства автоматизации, спроектировать архитектуру информационной системы и разработать пользовательский интерфейс</u>		
Краткие методические указания	<u>В ходе выполнения работы необходимо:</u>		
1.	Описать структуру разрабатываемого приложения в целом, выделить модули и конкретизировать задачи для каждого модуля (авторизация, работа с базой данных и т.д.), а также указать основные информационные объекты, которые используются в каждом модуле (модели данных, сервисы, и прочие подобные сущности фреймворка). Выбрать методологию и в соответствии с ее правилами сформировать набор диаграмм, дающих формальное описание. Сделать выводы о функциональных требованиях к средствам автоматизации со стороны смоделированных процессов. При наличии возможность описать нефункциональные требования		
2.	Описать типовые функциональные возможности классов информационных систем, применяющихся для автоматизации определенных на предыдущем этапе процессов, обосновать выбор конкретного набора информационных систем, детально описать их функциональные возможности и сопоставить их с функциональными требованиями, полученными на предыдущем этапе.		
3.	Представить функциональную и информационную архитектуры ИС, включающие все выбранные на предыдущем этапе программные средства автоматизации. Функциональная архитектура представляется как распределение операций смоделированных процессов по функциональным компонентам отдельных программных средств. В случае взаимосвязанных процессов или распределения операций одного процесса по нескольким средствам автоматизации указывается передача данных между функциональными компонентами соответствующих модулей. Информационная архитектура представляется в виде сопоставления информационных объектов, выделенных на первом этапе с информационными объектами, реализованными в выбранных средствах автоматизации. Описать интеграцию систем на уровне совместного использования преобразования данных информационных объектов, обеспечение целостности данных и синхронизации выполняемых над ними операций.		

Содержание пояснительной записки

1. Определение основных понятий
 2. Анализ и моделирование процессов
 3. Анализ средств автоматизации процессов
 4. Проектирование архитектуры ИС
 5. Реализация пользовательского интерфейса
-

Рекомендуемая литература

1. Мартин Фаулер - Архитектура корпоративных программных приложений. Издательский дом "Вильямс". 2006 г.
 2. Флэнаган, Дэвид. JavaScript. Полное руководство, 7-е изд. : Пер. с англ. — СПб. : ООО "Диалектика", 2021. — 720 с .
 3. Янг А., Мек Б., Кантелон М. Node.js в действии. 2-е изд. — СПб.: Питер, 2018. — 432 с.
 4. Браун И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. 2-е издание. — СПб.: Питер, 2021. — 336 с.
 5. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>. – Дата доступа: 04.05.2021.
-

Руководитель

Подпись, дата

Студент

Подпись, дата

УНИВЕРСИТЕТ ИТМО

АННОТАЦИЯ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

Студент Джахан Исрат
(Фамилия, И.О.)
Факультет Информационных технологий и программирования
Кафедра Информационных систем Группа М33091
Направление (специальность) 09.03.02 «Информационные системы и технологии»
Руководитель Андреев Николай Владимирович, Университет ИТМО, преподаватель
(квалификационная категория "преподаватель практики")
(Фамилия, И.О., место работы, должность, ученое звание, степень)
Дисциплина Web-программирование
Наименование темы Разработка веб-приложения "IMarket"

ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)

1. Цель и задачи работы ☐ Предложены студентом ☐ Сформулированы при участии студента ☐ Определены руководителем

Цель: Разработать веб-приложение.

Задачи:

- 1) Сформировать функциональные требования к веб-приложению.
- 2) Проанализировать функциональные возможности и выбрать набор средств автоматизации.
- 3) Сформировать функциональную и информационную архитектуру решения.

2. Характер работы

- ☐ Расчет ☐ Конструирование
☐ Моделирование ☐ Другое,

4. Содержание работы

- 1) Определение основных понятий
- 2) Анализ и моделирование процессов
- 3) Анализ средств автоматизации процессов
- 4) Проектирование архитектуры ИС
- 5) Реализация пользовательского интерфейса

5. Выводы

Студент _____

(подпись)

Руководитель _____

(подпись)

«_____» _____ 2024 г.

УНИВЕРСИТЕТ ИТМО

О Т З Ы В РУКОВОДИТЕЛЯ о выполнении курсового проекта (работы)

Студент Джахан Исрат

(Фамилия, И.О.)

Факультет Информационных технологий и программирования

Кафедра Информационных систем Группа М33091

Направление (специальность) 09.03.02 «Информационные системы и технологии»

Руководитель Андреев Николай Владимирович, Университет ИТМО, преподаватель
(квалификационная категория "преподаватель практики")

(Фамилия, И.О., место работы, должность, ученое звание, степень)

Дисциплина Web-программирование

Наименование темы Разработка веб-приложения "IMarket"

ОЦЕНКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)

№ п/п	Показатели	Оценка			
		5	4	3	0*
1.	Способность к работе с литературными источниками, справочной литературой, Интернет-ресурсами и т. п.				
2.	Использование иностранных источников				
3.	Способность к анализу и обобщению информационного материала				
4.	Владение базовыми знаниями в профессиональной области				
5.	Владение базовыми знаниями в смежных областях				
6.	Владение навыками решения технических задач				
7.	Способность применять знания на практике				
8.	Уровень и корректность использования в работе методов численного моделирования, инженерных расчетов и статистической обработки данных				
9.	Владение навыками использования современных пакетов компьютерных программ и технологий				
10.	Владение навыками оформления отчетных материалов с применением современных пакетов программ				
11.	Качество оформления пояснительной записки (общий уровень грамотности, стиль изложения, качество иллюстраций, корректность цитирования и пр.**)				
12.	Качество оформления презентации				
13.	Владение навыками публичного выступления и межперсональной коммуникации				
14.	Владение навыками планирования и управления временем при выполнении работы				
ИТОГОВАЯ ОЦЕНКА					

* - не оценивается (трудно оценить)

**Отмеченные
достоинства:** _____

**Отмеченные
недостатки:** _____

Заключение _____

Руководитель _____

Дата « _____ » _____ 2024 г.

(подпись)

Оглавление

Введение.....	2
Определения, обобщения и сокращения	3
Описание предметной области	4
Описание прикладного процесса.....	4
Формирование требований	4
Проектирование.....	5
Используемый стек технологий	5
Системная архитектура	5
Архитектура данных.....	6
Программная архитектура.....	7
Разработка	22
Реализация серверного API.....	22
Реализация пользовательского интерфейса	23
Заключение	25
Список использованной литературы.....	26

Введение

Объектом разработки является интернет-магазин на языке программирования TypeScript с использованием фреймворка Nest – для серверной части приложения и языка программирования JavaScript, без использования фреймворков – для клиентской части приложения.

Выбранный фреймворк один из самых поддерживаемых и распространённых.

Целью работы является разработка веб-приложения и пользовательского интерфейса, анализ требования и моделирование процессов, средств автоматизации и архитектуры информационной системы.

В ходе работы были получены следующие результаты:

- Серверная часть системы, принимающая запросы.
- Клиентская часть системы, предоставляющая интерфейс пользователя.
- База данных для хранения информации о пользователях, файлах, а также служебной информации внутри системы.

Определения, обобщения и сокращения

Браузер (Browser) — это специальная программа, которая позволяет пользователям искать информацию в Интернете, просматривать веб-сайты, скачивать файлы различных форматов и воспроизводить аудио и видео файлы. Браузер является посредником между пользователем и Интернетом.

Фреймворк (Framework) — это готовая модель, шаблон или каркас, предназначенный для разработки программного обеспечения. Он предоставляет основные инструменты, библиотеки и структуру, которые помогают разработчикам создавать приложения, необходимые для определенной платформы или задачи.

API (Application Programming Interface) — это контракт, определяющий способ взаимодействия между различными программными компонентами. API предоставляет набор функций, процедур, классов и протоколов, которые разработчики могут использовать для создания приложений.

TypeORM — это **ORM (Object-Relational Mapping)** нового поколения с открытым исходным кодом для Node.js и TypeScript. TypeORM предоставляет инструменты для работы с базами данных в объектно-ориентированном стиле, позволяя разработчикам взаимодействовать с данными с использованием объектов и методов, вместо написания прямых SQL-запросов. TypeORM автоматически выполняет маппинг объектов на таблицы базы данных и обеспечивает типобезопасность для кода.

Supertokens — это инструмент для управления аутентификацией и авторизацией в веб-приложениях. Он предоставляет готовые решения для функций, связанных с безопасностью пользователей, таких как регистрация, вход, сброс пароля и многое другое. Supertokens упрощает процесс реализации безопасности веб-приложений, предоставляя API и компоненты, которые можно использовать для управления пользовательскими данными и сессиями.

Приведенные выше определения и инструменты являются важными для разработки приложений и создания удобного пользовательского опыта в современном веб-разработке.

Описание предметной области

Описание прикладного процесса

Разрабатываемое веб-приложение представляет собой интернет-магазин, позволяющий управлять, добавлять, удалять, редактировать и представлять продукты.

Пользователи могут зарегистрироваться в системе, предоставив необходимую информацию, такую как имя, адрес электронной почты и пароль.

Зарегистрированные пользователи могут войти в систему, используя свои учетные данные.

Формирование требований

В ходе анализа прикладного процесса был получен следующий список функциональных требований:

Модуль работы с разделами должен иметь:

- Возможность зарегистрироваться, авторизоваться
- Возможность к чату в реальном времени

Нефункциональные требования

Разрабатываемая система не является публичной, поэтому основным требованием выступает её закрытость, путем обязательного прохождения процесса авторизации и аутентификации.

Так же должна быть возможность работы с приложением напрямую, через общие программные интерфейсы, описанные по спецификации OpenAPI версии не ниже 3.0

Проектирование

Используемый стек технологий

Решение создания именно веб-приложения обусловлено тем, что необходимо было обеспечить доступ к системе с любого устройства, в любое время. Веб-приложение решает этот вопрос, а также снимает вопрос обновлений на стороне клиента.

Проект использует стек стандартных технологий, характерный для большинства веб-приложений: HTML, CSS, Javascript.

В качестве веб-сервера используется Nest.

В качестве базы данных используется PostgreSQL, ввиду того, что данная СУБД является самой стабильной в данной связке. Библиотекой для работы с данными была выбрана TypeORM.

В проекте используется свободная распределённая система управления версиями Git, хранилищем исходных кодов является крупнейший веб-сервис GitHub, а в качестве хостинга используется облачный сервис Render.com.

Системная архитектура

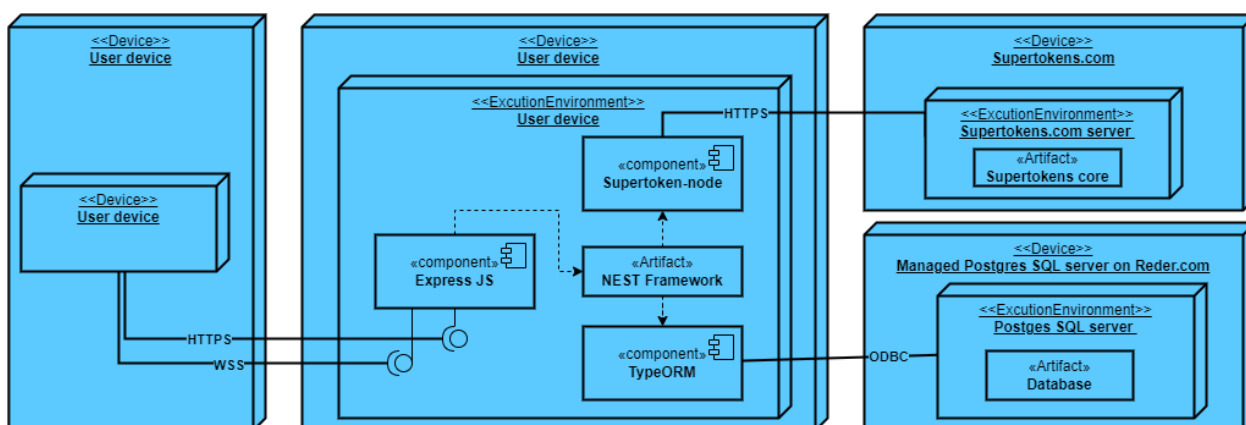


Рисунок 1. Системная архитектура приложения.

Архитектура данных

Представления схемы:

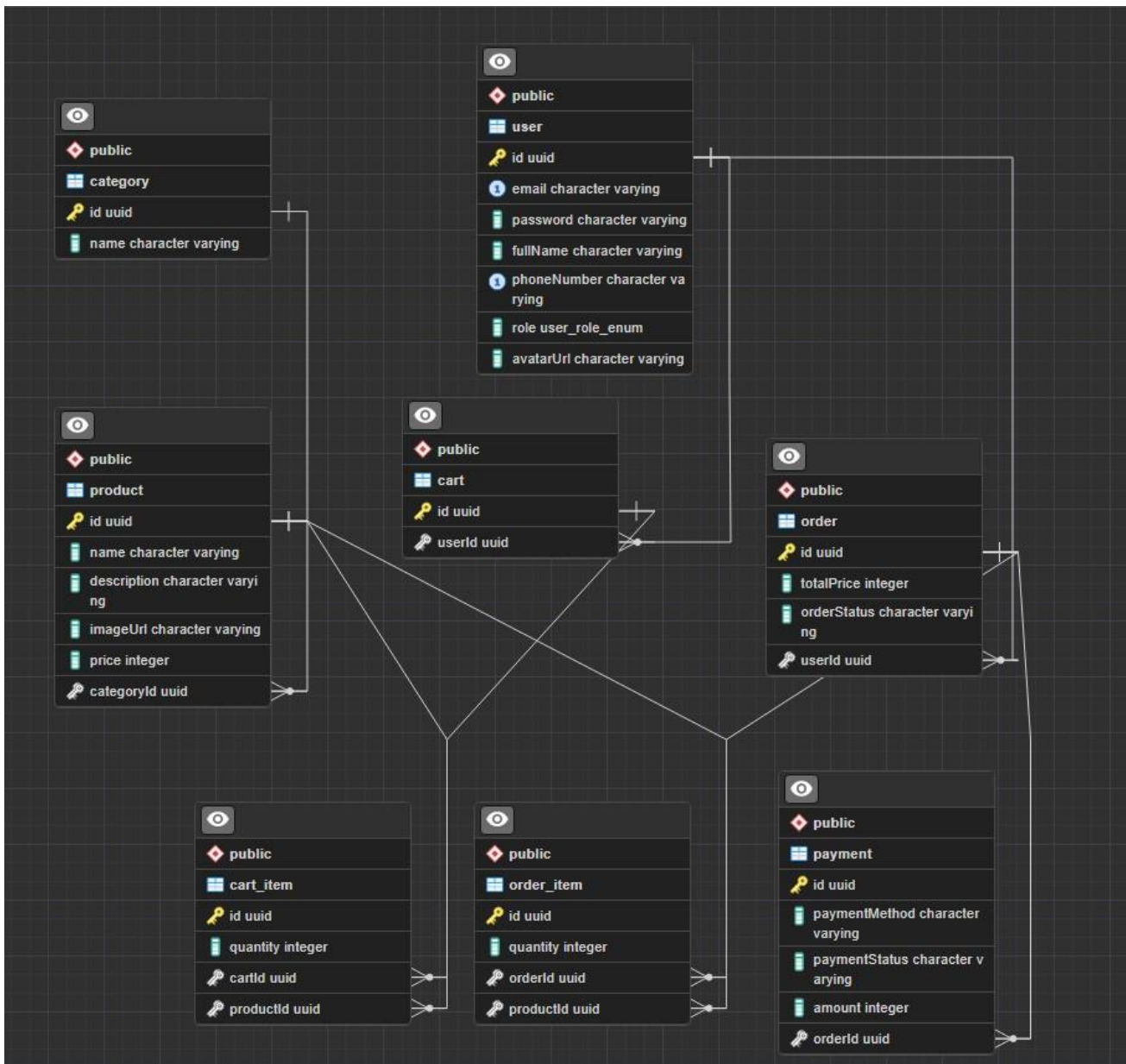


Рисунок 2. Схема таблиц базы данных

Relationships:

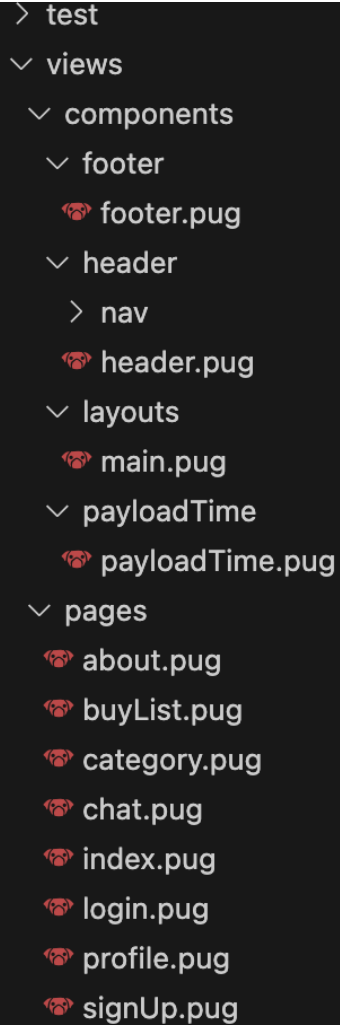
- User - Order: One-to-Many
- User - Cart: One-to-One
- Product - Category: Many-to-One
- Product - CartItem: One-to-Many
- Product - OrderItem: One-to-Many
- Order - OrderItem: One-to-Many

- Cart - CartItem: One-to-Many
- Order - Payment: One-to-One

Программная архитектура

1. Frontend

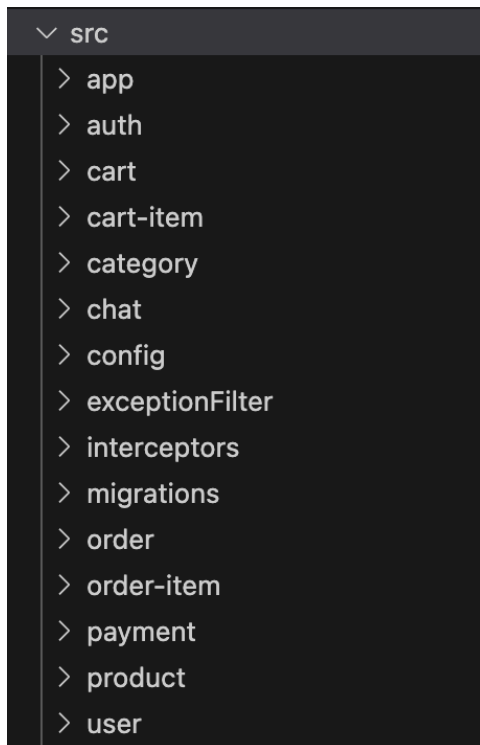
Я использую шаблонизатор “Pug”. Части (partials) веб страниц к выделению в отдельные представления.



```
> test
  ✓ views
    ✓ components
      ✓ footer
        🐶 footer.pug
      ✓ header
        > nav
        🐶 header.pug
    ✓ layouts
      🐶 main.pug
    ✓ payloadTime
      🐶 payloadTime.pug
    ✓ pages
      🐶 about.pug
      🐶 buyList.pug
      🐶 category.pug
      🐶 chat.pug
      🐶 index.pug
      🐶 login.pug
      🐶 profile.pug
      🐶 signUp.pug
```

2. Backend

В работе находятся следующие модули:



User:

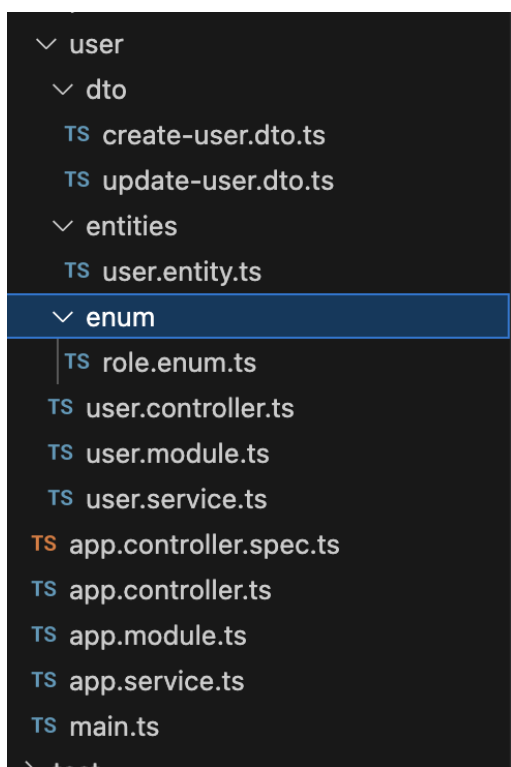


Таблица 1. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
UserModule	User	Модель
	CreateUserDto	Модель передачи данных
	UpdateUserDto	Модель передачи данных
	UsersService	Сервис
	UsersController	Сервис (Контроллер)
	UsersModule	Модуль

Таблица 2. Описание классов

Название класса	Описание класса
User	Класс представляет пользователя
CreateUserDto	Модель передачи данных в сервис для создания пользователей
UpdateUserDto	Модель передачи данных в сервис для обновления пользователей
UsersService	Класс UsersService используется для работы с пользователями. Он обеспечивает
	функциональность по созданию, чтению, обновлению и удалению пользователей, а также выполнению других операций связанных с базой данных.
UsersController	Класс UsersController является контроллером, отвечающим за обработку HTTP-запросов, связанных с пользователями. Он содержит несколько методов для обработки различных операций, таких как создание, чтение, обновление и удаление пользователей.
UsersModule	Класс UsersModule представляет модуль NestJS, который объединяет компоненты, связанные с пользователями, и определяет их зависимости и конфигурацию.

Auth:

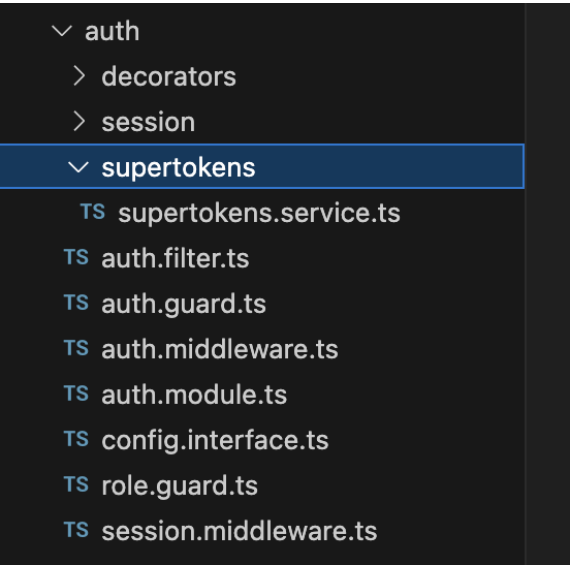


Таблица 3. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
AuthModule	SupertokensService	Сервис
	SupertokensExceptionHandler	Фильтры исключений
	AuthGuard	Защитники/предохранители
	AuthMiddleware	Посредники
	AuthModule	Модуль

Таблица 4. Описание классов

Название класса	Описание класса
SupertokensService	Класс SupertokensService представляет сервис для инициализации и настройки Supertokens, который является инструментом для обеспечения безопасности и аутентификации веб-приложений.

SupertokensExceptionHandler	Класс SupertokensExceptionHandler является пользовательским фильтром исключений в NestJS. Он реализует интерфейс ExceptionFilter из NestJS, что позволяет обрабатывать исключения, возникшие в приложении.
AuthGuard	Класс AuthGuard является пользовательским гвардом (guard) в NestJS. Он реализует интерфейс CanActivate, который используется для выполнения проверок доступа перед обработкой запроса
AuthMiddleware	Класс AuthMiddleware является Middleware в NestJS для обработки аутентификации с использованием библиотеки Supertokens. Он обеспечивает промежуточную обработку запросов, связанных с аутентификацией, и передает их в Supertokens для выполнения необходимых действий.
AuthModule	Класс AuthModule используется для организации функциональности аутентификации.

Product:

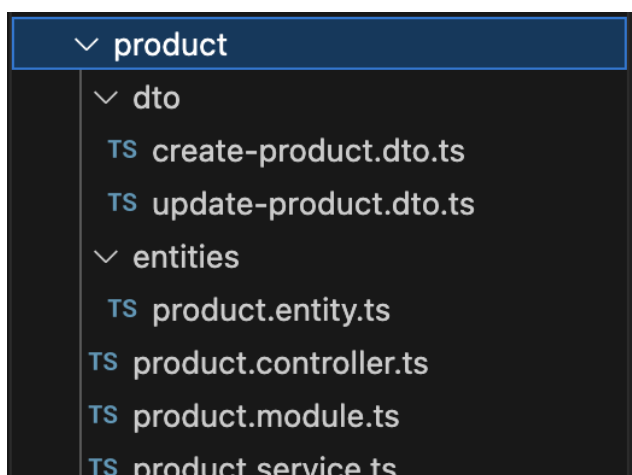


Таблица 5. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
ProductModule	Product	Модель
	ProductCreateDto	Модель передачи данных
	Product UpdateDto	Модель передачи данных
	ProductService	Сервис
	ProductController	Сервис (Контроллер)
	Product Module	Модуль

Таблица 6. Описание классов

Название класса	Описание класса
Product	Класс Boot представляет сущность "бутс" и используется для работы с данными о ботинках в системе. Он отображается в базе данных в виде таблицы, где каждая строка представляет отдельный ботинок.
Product CreateDto	Модель передачи данных в сервис для создания бутса
Product UpdateDto	Модель передачи данных в сервис для обновления бутса
Product Service	Класс Product Service представляет сервис, который используется для работы с данными о бутсах. Он обеспечивает функциональность по созданию, чтению, обновлению и удалению бутсов.

Таблица 7. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
UserModule	Cart	Модель
	CreateCartDto	Модель передачи данных
	UpdateCartDto	Модель передачи данных
	CartService	Сервис
	CartController	Сервис (Контроллер)
	CartModule	Модуль
	CartItem	Модель
	CreateCartItemDto	Модель передачи данных
	UpdateCartItemDto	Модель передачи данных
	CartItemsService	Сервис
	CartItemsController	Сервис (Контроллер)
	CartItemsModule	Модуль

Таблица 8. Описание классов

Название класса	Описание класса
Cart	Класс Cart представляет сущность корзины в приложении. Он связан с сущностью User (пользователь) с помощью отношения один к одному (OneToOne).
CreateCartDto	Модель передачи данных в сервис для создания корзины
UpdateCartDto	Модель передачи данных в сервис для обновления корзины
CartsService	Класс CartsService представляет сервис для работы с сущностью корзины (Cart). Он содержит методы для создания, поиска, обновления и удаления корзины, а

	также для сохранения изменений.
CartsController	Класс CartsController представляет контроллер для обработки HTTP-запросов, связанных с корзинами (Cart). Он содержит методы для создания, получения, обновления и удаления корзин.
CartsModule	Модуль CartsModule является модулем Nest.js, который объединяет сервисы, контроллеры и другие зависимости, связанные с функциональностью корзин (Cart).
CartItem	Класс CartItem представляет сущность элемента корзины (CartItem). Он содержит информацию о товаре (Boot), количестве и связи с корзиной (Cart).
CreateCartItemDto	Модель передачи данных в сервис для создания товаров в корзине
UpdateCartItemDto	Модель передачи данных в сервис для обновления товаров в корзине
CartItemsService	Класс CartItemsService представляет сервис для работы с элементами корзины (CartItem). Он содержит методы для создания, получения, обновления и удаления элементов корзины.

CartItemsController	Класс CartItemsController представляет контроллер для обработки запросов, связанных с элементами корзины (CartItem).
CartItemsModule	Класс CartItemsModule представляет собой модуль в NestJS, используемый для работы с элементами корзины. Он предоставляет зависимости, контроллеры и сервисы, связанные с элементами корзины.

Order:

```

  ▾ order
    ▾ dto
      TS create-order.dto.ts
      TS update-order.dto.ts
    ▾ entities
      TS order.entity.ts
    TS order.controller.ts
    TS order.module.ts
    TS order.service.ts
    TS orderStatus.enum.ts
  ▾ order-item
    ▾ dto
      TS create-order-item.dto.ts
      TS update-order-item.dto.ts
    ▾ entities
      TS order-item.entity.ts
    TS order-item.controller.ts
    TS order-item.module.ts
    TS order-item.service.ts

```

Таблица 9. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
OrderModule	Order	Модель
	CreateOrderDto	Модель передачи данных
	Update Order Dto	Модель передачи данных
	OrdersService	Сервис
	OrdersController	Сервис (Контроллер)
	OrdersModule	Модуль
	OrderItem	Модель

	CreateOrderItemDto	Модель передачи данных
	UpdateOrderItemDto	Модель передачи данных
	OrderItemsService	Сервис
	OrderItemsController	Сервис (Контроллер)
	OrderItemsModule	Модуль

Таблица 10. Описание классов

Название класса	Описание класса
Order	Класс Order является сущностью в NestJS, используемой для работы с заказами. Он содержит информацию о заказе, такую как идентификатор, пользователя, дату создания, платежную информацию, статус заказа и общую сумму.
CreateOrderDto	Модель передачи данных в сервис для создания заказа
Update Order Dto	Модель передачи данных в сервис для обновления заказа
OrdersService	Класс OrdersService является сервисом в NestJS, используемым для работы с заказами. Он содержит методы для создания, получения, обновления и удаления заказов. Также сервис взаимодействует с другими сервисами и репозиторием для выполнения необходимых операций.
OrdersController	Класс OrdersController является контроллером в NestJS, используемым для обработки запросов, связанных с заказами. Он содержит методы для создания, получения, обновления и удаления заказов.

OrdersModule	Класс OrdersModule является модулем NestJS, который используется для управления зависимостями и организации компонентов, связанных с заказами.
OrderItem	Класс OrderItem используется для работы с элементами заказа. Помогает искать объекты, отображает полный список позиций заказа и предоставляет возможность удалить позицию заказа из общего списка.
CreateOrderItemDto	Модель передачи данных в сервис для создания товара в заказе
UpdateOrderItemDto	Модель передачи данных в сервис для обновления товара в заказе
OrderItemsService	Класс OrderItemService — это класс обслуживания, используемый для работы с элементами заказа. Он предоставляет методы для создания, чтения, обновления и удаления элементов заказа, а также их сохранения в базе данных.
OrderItemsController	Класс OrderItemController — это контроллер, используемый для обработки запросов, связанных с элементами заказа. Он предоставляет маршруты и методы для выполнения таких операций, как поиск, обновление и удаление элементов заказа.
OrderItemsModule	Класс OrderItemModule — это модуль в приложении NestJS, который используется для управления элементами заказа.

Payment:

```

  ▾ payment
    ▾ dto
      TS create-payment.dto.ts
      TS update-payment.dto.ts
    ▾ entities
      TS payment.entity.ts
    ▾ enum
      TS paymentMethod.enum.ts
      TS paymentStatus.enum.ts
    TS payment.controller.ts
    TS payment.module.ts
    TS payment.service.ts

```

Таблица 11. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
Payment Module	Payment	Модель
	CreatePaymentDto	Модель передачи данных
	UpdatePaymentDto	Модель передачи данных
	PaymentsService	Сервис
	PaymentsController	Сервис (Контроллер)
	PaymentsModule	Модуль

Таблица 12. Описание классов

Название класса	Описание класса
Payment	это сущность в приложении. Он представляет информацию о доставке в системе.
CreatePaymentDto	Модель передачи данных в сервис для создания оплаты
UpdatePaymentDto	Модель передачи данных в сервис для обновления оплаты
PaymentsService	Класс Payment Service — это служба (служба) в приложении, используемая для манипулирования платежной информацией.
PaymentsController	Класс PaymentsController используется для работы с платежами. Он предоставляет возможность создавать новые платежи, получать список всех платежей, находить платеж по идентификатору и обновлять или удалять платежи из общего каталога.
PaymentsModule	Модуль PaymentsModule определяет зависимости и настройки для работы с платежами в приложении.

Разработка

Реализация серверного API

В качестве описания программного интерфейса был выбран инструмент, поддерживающий стандарт OAS 3.0 – Swagger. Далее представлена полученная документация API полученная автоматически по директивам, указанным в декораторах различных методов и структурах данных внутри разрабатываемой информационной системы.

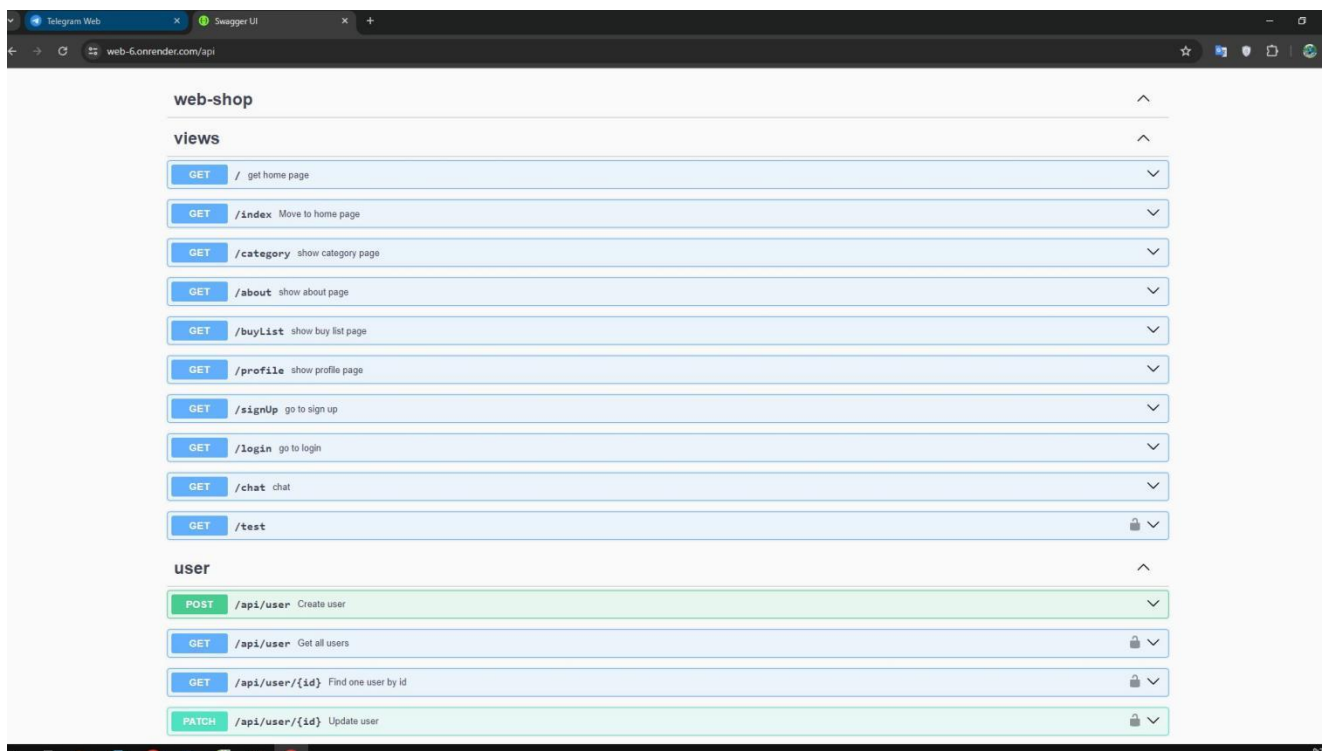



Рисунок N. Программный интерфейс серверного API.

Реализация пользовательского интерфейса


представления продукта



Главная Категории О нас Список покупок Chat

Login Sign Up


Total load time: 113625 ms (client) and 2 ms (server)



Samsung Galaxy Tab S8

87,499pp


Купить



Программирование на Javascript

540pp

Купить




iPad Pro 4

220,000pp

Купить

Смартфоны

Регистрация



Главная Категории О нас Список покупок Chat

Login Sign Up

Sign Up

Email:

Password:


Full name:

Phone number:

Submit

© 2023 Israt Market. Все права защищены.

Авторизоваться



ГлавнаяКатегориюО насСписок покупкиChat

LoginSign Up

LOGIN


Email:

Password

Login

© 2023 Israt Market. Все права защищены.

Чат в реальном времени



ГлавнаяКатегориюО насСписок покупкиChat

LoginSign Up

Chat

Enter message...

Send

2024-05-03T06:16:01.000Z
Anonymous
p;mnzcszsfhkilej:>OCa]q[oi'pHCF;o

2024-05-03T06:15:53.496Z
Anonymous
,kahdkqyrflika.jcnf.a

2024-05-03T06:15:50.287Z
Anonymous
,mdk.ajfl

2024-04-15T11:31:08.578Z
Anonymous
Hello!

2024-04-15T11:30:50.336Z
Anonymous

Заключение

В ходе выполнения курсовой работы был проведён анализ работы Разработка веб-приложения "IMarket" , исходя из которого были выявлены и сформированы требования к разрабатываемому веб-приложению.

Исходя из выбранной архитектуры и наложенных ограничений были сформированы требования к используемым технологиям внутри модулей.

Была спроектирована архитектура данных, программная и системная архитектура в виде набора диаграмм в нотации UML.

Опираясь на вышеизложенные требования и стек технологий было разработано веб-приложение и пользовательский интерфейс в рамках дисциплины «Web-программирование».

Таким образом, все поставленные ранее цели были выполнены.

Разработанное приложение является результатом данной курсовой работы.

Список использованной литературы

1. Мартин Фаулер - Архитектура корпоративных программных приложений. Издательский дом "Вильямс". 2006 г.
2. Флэнаган, Дэвид. JavaScript. Полное руководство, 7-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2021. — 720 с .
3. Янг А., Мек Б., Кантелон М. Node.js в действии. 2-е изд. — СПб.: Питер, 2018. — 432 с.
4. Браун И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. 2-е издание. — СПб.: Питер, 2021. — 336 с.
5. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>. – Дата доступа: 04.05.2021.