

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**УНИВЕРСИТЕТ ИТМО**

---

**Кафедра информационных систем**

**КУРСОВАЯ РАБОТА**

**Тема: Разработка веб-приложения «Сайт-портфолио с лентой коротких постов»**

**Работу выполнил студент:** Гуммель Никита Константинович группы М33041  
(фамилия, имя, отчество) (номер группы)

**Руководитель** Плохотнюк Вадим Станиславович, *преподаватель практики*  
(фамилия, имя, отчество)

**Работа защищена** " \_\_\_\_\_ " \_\_\_\_\_ 2022 г. с оценкой \_\_\_\_\_

**Подписи членов комиссии:** \_\_\_\_\_

\_\_\_\_\_

**САНКТ-ПЕТЕРБУРГ  
2022**

## УНИВЕРСИТЕТ ИТМО

### ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)

Студент	<u>Гуммель Никита Константинович</u> ( Фамилия, И., О. )		
Факультет	<u>Информационных технологий и программирования</u>		
Кафедра	<u>Информационных систем</u>	Группа	<u>М33041</u>
Направление (специальность)	<u>Информационные системы и технологии</u>		
Руководитель	<u>Плохотнюк Вадим Станиславович, Университет ИТМО, преподаватель практики.</u> ( Фамилия, И.О., должность, ученое звание, степень )		
Дисциплина	<u>Web-программирование</u>		
Наименование темы	<u>Разработка веб-приложения «Сайт-портфолио с лентой коротких постов»</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 Гуммель Никита Константинович

(Фамилия, И.О.)

Факультет Информационных технологий и программирования

Кафедра Информационных систем Группа М33041

Направление (специальность) 09.03.02 «Информационные системы и технологии»

Руководитель Плохотнюк Вадим Станиславович, Университет ИТМО, преподаватель практики.

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

Дисциплина Web-программирование

Наименование темы Разработка веб-приложения «Сайт-портфолио с лентой коротких постов»

### 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)

« \_\_\_\_\_ » \_\_\_\_\_ 2022 г.

# УНИВЕРСИТЕТ ИТМО

## О Т З Ы В РУКОВОДИТЕЛЯ

### о выполнении курсового проекта (работы)

Студент Гуммель Никита Константинович  
( Фамилия, И.О.)

Факультет Информационных технологий и программирования

Кафедра Информационных систем Группа М33041

Направление (специальность) 09.03.02 «Информационные системы и технологии»

Руководитель Плохотнюк Вадим Станиславович, Университет ИТМО,  
преподаватель практики  
( Фамилия, И.О., место работы, должность, ученое звание, степень )

Дисциплина Web-программирование

Наименование темы Разработка веб-приложения «Сайт-портфолио с лентой коротких постов»

### 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)

**ДОСТОИНСТВА:** \_\_\_\_\_

[illegible]

**недостатки:** \_\_\_\_\_

[illegible]

## Заключение

---

---

---

---

---

---

(подпись)

Дата « \_\_\_\_\_ » \_\_\_\_\_ 2022 г.

## Оглавление

Введение.....	2
Определения, обобщения и сокращения .....	3
Описание предметной области .....	5
Описание прикладного процесса.....	5
Формирование требований .....	5
Проектирование.....	6
Используемый стек технологий .....	6
Системная архитектура .....	6
Архитектура данных.....	7
Программная архитектура.....	8
Разработка .....	10
Реализация серверного API.....	10
Реализация пользовательского интерфейса .....	11
Заключение .....	14
Список использованной литературы .....	15

## Введение

Объектом разработки является сайт-портфолио с лентой коротких постов 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* – прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов

*Фреймворк* – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

*Express* – фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT.

*Prisma* – это инструмент, позволяющий работать с реляционными и нереляционными базами данных с помощью JavaScript или TypeScript без использования SQL.

*PostgreSQL* – свободная объектно-реляционная система управления базами данных.

*UML* – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

*API* – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

*Авторизация* – предоставление определённого лицу или группе лиц прав на выполнение определённых действий.

Аутентификация – это проверка подлинности лица, которое хочет получить доступ к системе.

*TCP/IP* – сетевая модель передачи данных, представленных в цифровом виде. Модель описывает способ передачи данных от источника информации к получателю.

*URL* – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса. Используется как стандарт записи ссылок на объекты в Интернете.

## **Описание предметной области**

### **Описание прикладного процесса**

Разрабатываемое веб-приложение, это сайт-портфолио с лентой коротких постов. В ленте есть возможность просматривать выкладываемые людьми изображения с подписями или только заметки, а также публиковать данные короткие посты авторизованными пользователями.

В данном проекте предусмотрен механизм просмотра ленты, авторизации/регистрации, публикации коротких постов.

### **Формирование требований**

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

Модуль работы с пользователями должен иметь:

- Регистрация/авторизация через почту и OTP код

Модуль работы с постами должен иметь:

- Публикация постов
- Получение ленты из незабаненных постов
- Уведомление о выходе нового поста в ленте

### **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 Express.

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

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

## Системная архитектура

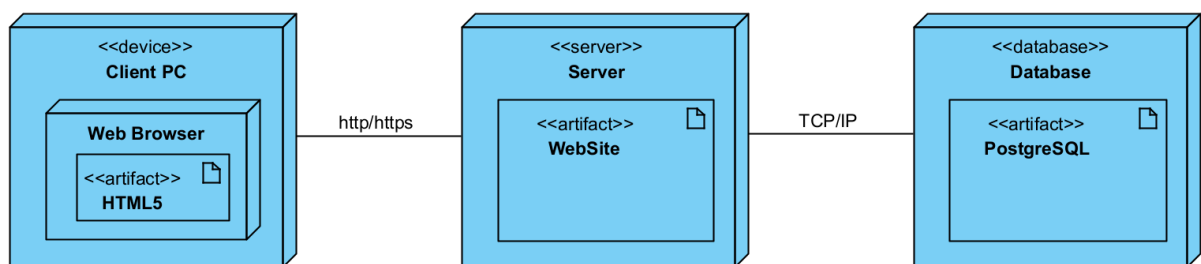


Рисунок 1. Системная архитектура приложения.

На рисунке 1 изображена UML диаграмма системной архитектуры приложения. Клиент с помощью браузера взаимодействует с сервером через протокол http или https, сервер взаимодействует с базой данных PostgreSQL через протокол TCP/IP.

## Архитектура данных

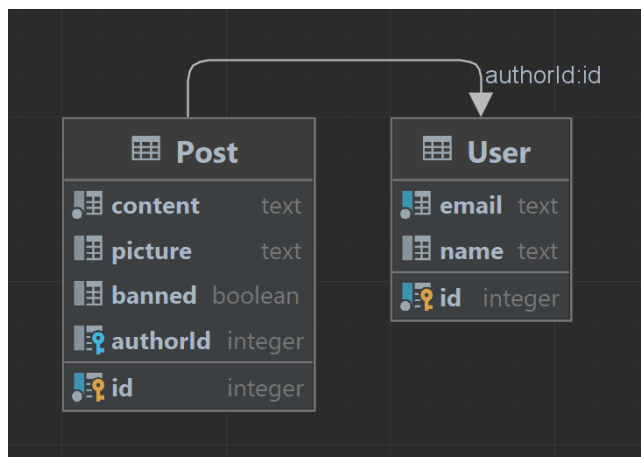


Рисунок 2. Схема таблиц базы данных

User – Предоставляет информацию о пользователе.

- email – почта, уникально идентифицирующая пользователя.
- name – имя пользователя для отображения в постах.
- id – идентификатор пользователя в системе.

Post – Предоставляет информацию о посте.

- content – текстовое содержимое поста.
- picture – опциональное поле, ссылка на изображение, которое будет отображено в посте.
- banned – статус поста, если true – пост не соответствует правилам платформы и не будет показываться в ленте.
- authorId – идентификатор пользователя, опубликовавшего пост.
- id – идентификатор поста в системе.

## Программная архитектура

Таблица 1. Отношения модулей и классов

Название модуля	Название класса	Назначение класса
auth	supertokens.service	Сервис (Модель)
	auth.filter	Обработка ошибок
	auth.guard	Проверка сеанса
	auth.middleware	Фильтрация данных
	auth.module	Корневая точка модуля
	config.interface	Конфигурация SuperToken
	session.decorator	Получение сессии
post	post.controller	Сервис (Контроллер)
	post.module	Корневая точка модуля
	post.service	Сервис (Модель)
	post.dto	Модель передачи данных
user	user.controller	Сервис (Контроллер)
	user.module	Корневая точка модуля
	user.service	Сервис (Модель)
	user.dto	Модель передачи данных
widget	widget.gateway	Класс-обёртка
	widget.controller	Сервис (Контроллер)
	widget.module	Корневая точка модуля

Таблица 2. Описание классов

Название класса	Описание класса
supertokens.service	Класс, реализующий методы данного модуля.
auth.filter	Класс, реализующий обработку ошибок авторизации.
auth.guard	Класс, реализующий проверку подлинности сессии.
auth.middleware	Класс, реализующий проверку и фильтрацию поступающих данных на авторизацию.

auth.module	Класс, используемый для организации внутренней структуры фреймворком Nest.
config.interface	Класс, отвечающий за тип и токен инъекции для конфигурации SuperToken
session.decorator	Класс-декоратор, возвращающий сеанс пользователя.
post.controller	Класс, который занимается обработкой входящих запросов и возвращением ответов для работы с постами.
post.module	Класс, используемый для организации внутренней структуры фреймворком Nest.
post.service	Класс, реализующий методы модуля постов.
post.dto	Класс, инкапсулирующий и валидирующий данные при передаче из одной подсистемы в другую.
user.controller	Класс, который занимается обработкой входящих запросов и возвращением ответов для работы с пользователями.
user.module	Класс, используемый для организации внутренней структуры фреймворком Nest.
user.service	Класс, реализующий методы модуля пользователей.
user.dto	Класс, инкапсулирующий и валидирующий данные при передаче из одной подсистемы в другую.
widget.gateway	Класс, обслуживающий WebSocket.
widget.controller	Класс, который занимается обработкой входящих запросов и возвращением ответов при работе с WebSocket.
widget.module	Класс, используемый для организации внутренней структуры фреймворком Nest.

# 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.

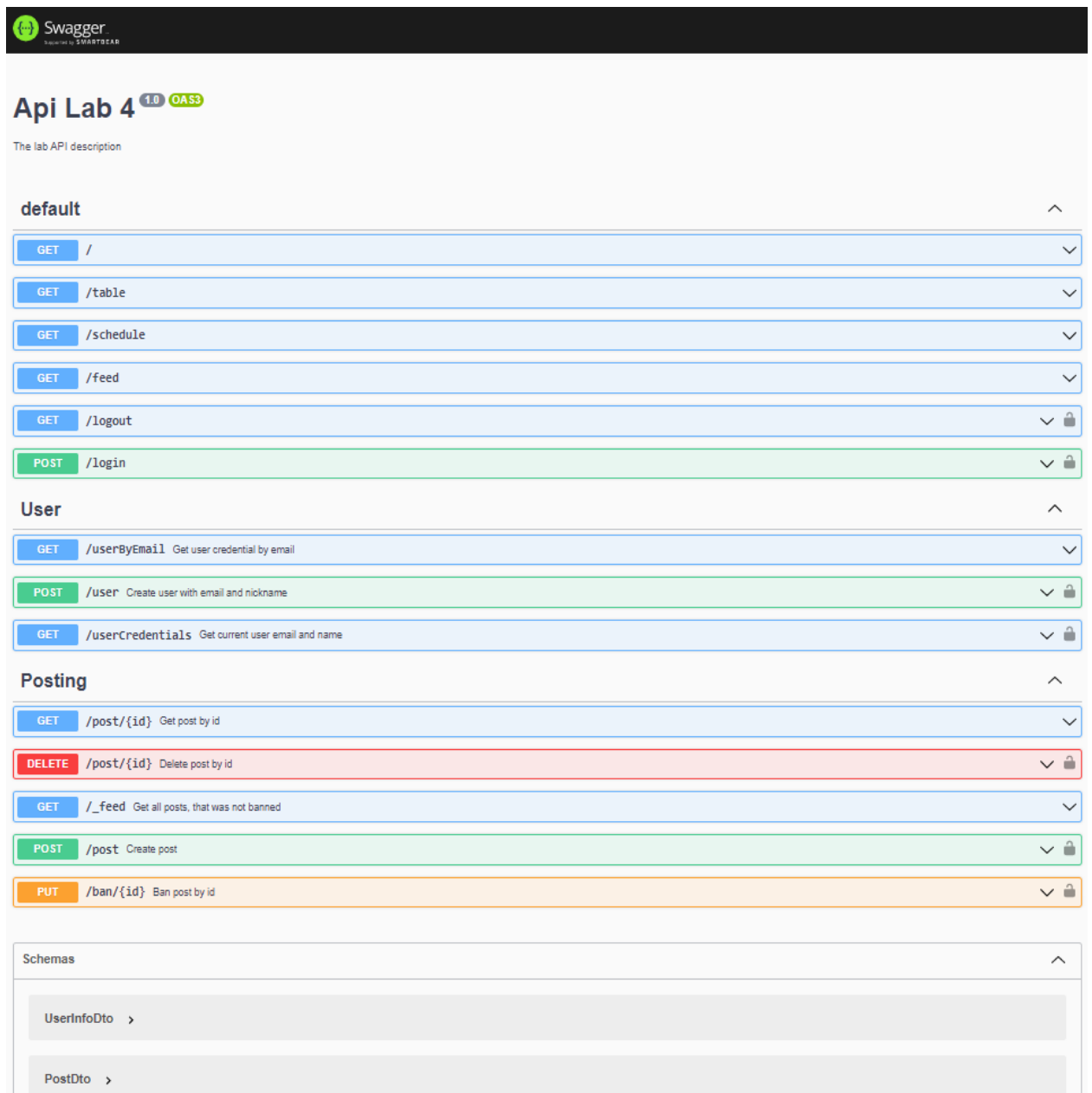


Рисунок 3. Программный интерфейс серверного API.



## Реализация пользовательского интерфейса

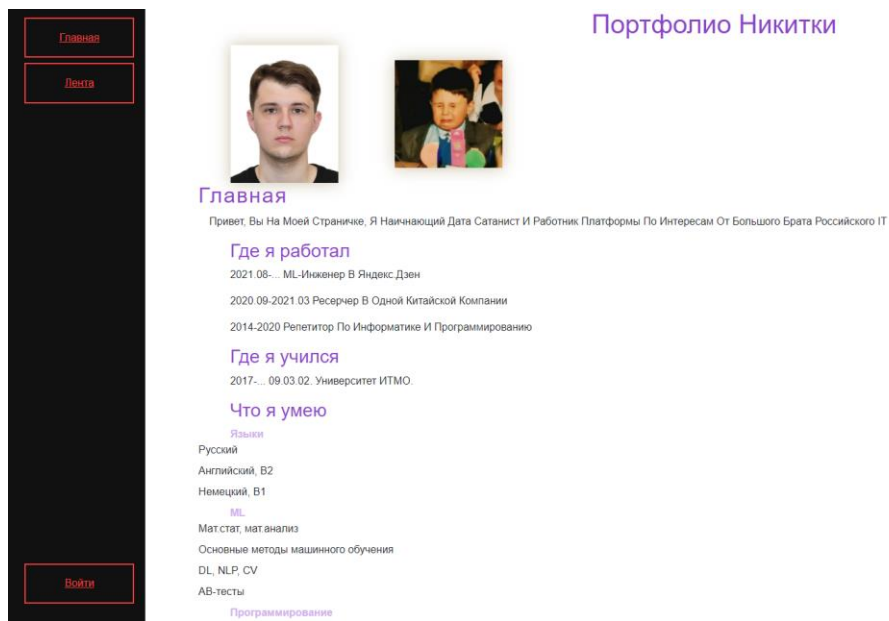


Рисунок 4.1. Отображение главной страницы.

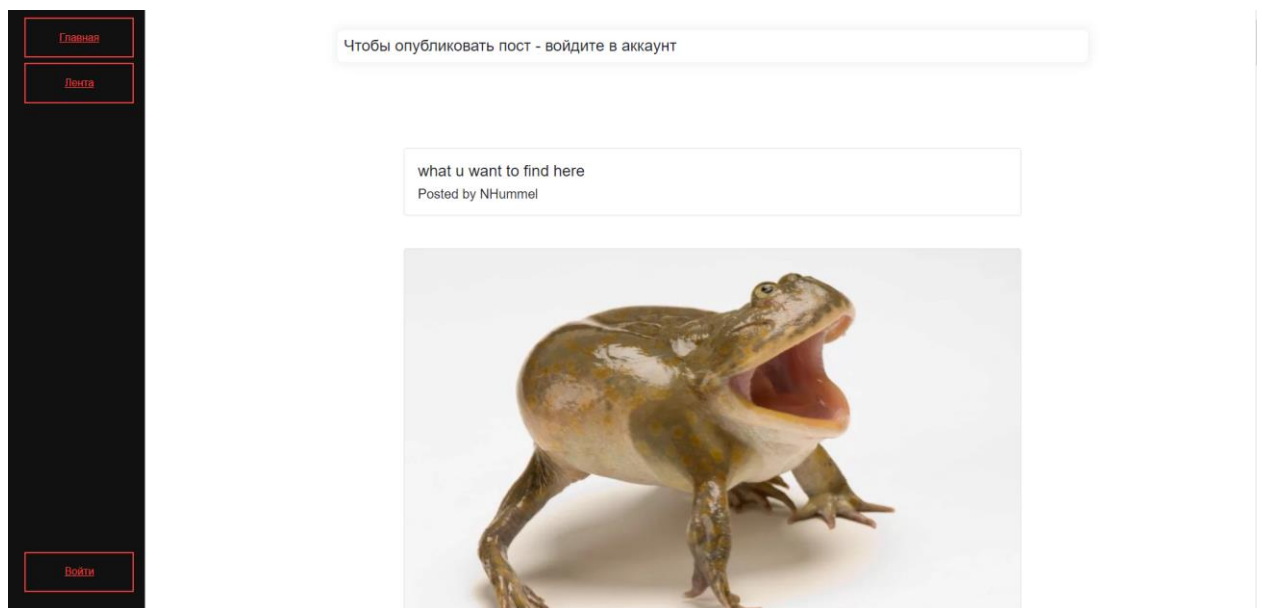


Рисунок 4.2.1. Отображение ленты (анонимный пользователь).

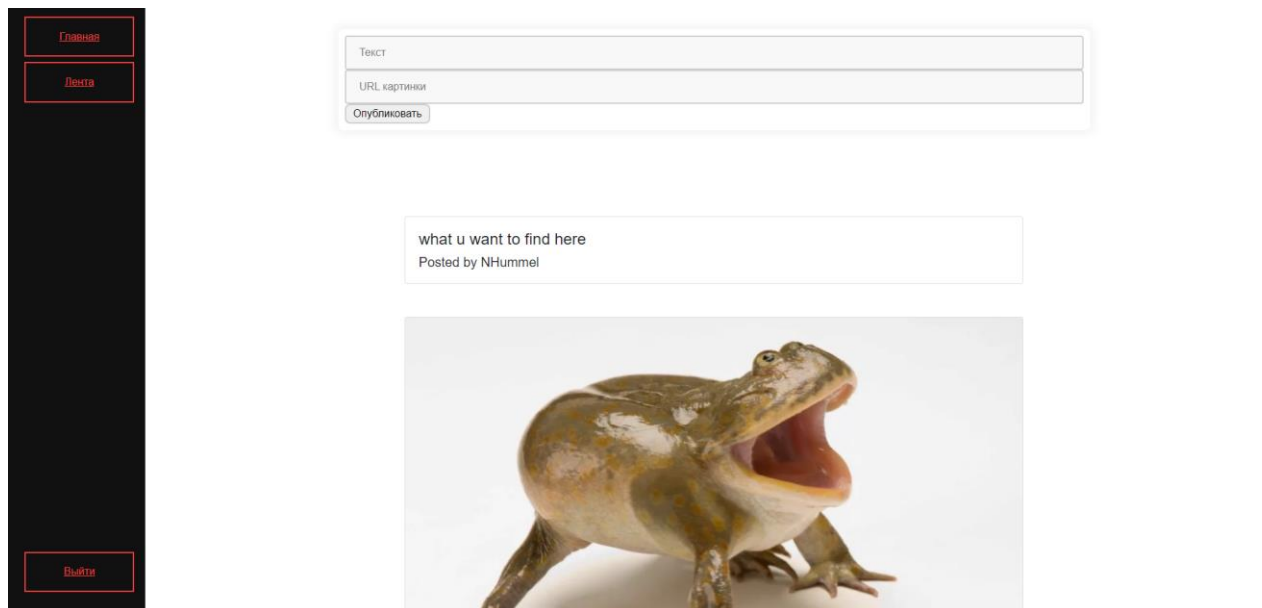


Рисунок 4.2.2. Отображение ленты (авторизованный пользователь).

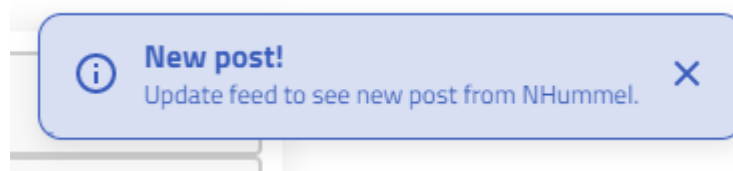
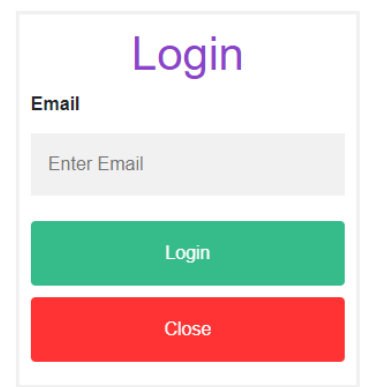
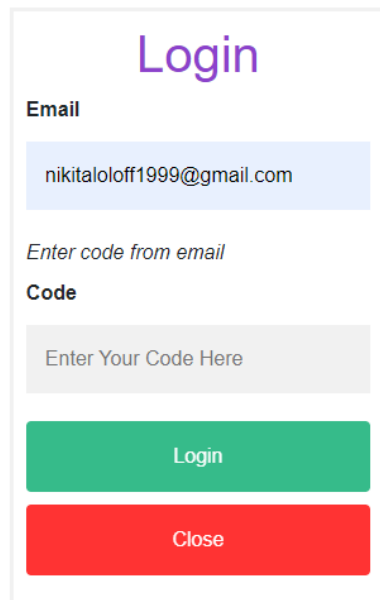


Рисунок 4.3. Нотификация о выходе нового поста, реализованная через WebSocket.



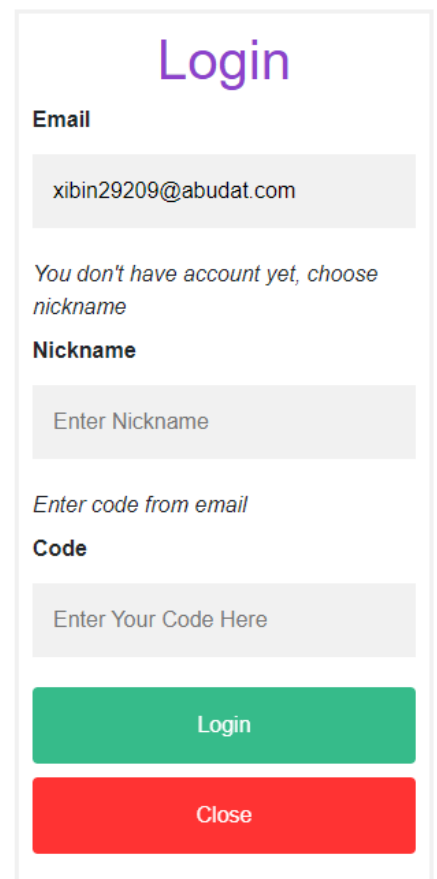
A login form titled "Login" in purple. It features an "Email" label above a light gray input field containing the placeholder text "Enter Email". Below the input field are two buttons: a green "Login" button and a red "Close" button.

Рисунок 4.4.1.  
Окно авторизации  
(ввод почты)



A login form titled "Login" in purple. It features an "Email" label above a light blue input field containing the email "nikitaloloff1999@gmail.com". Below the input field is the text "Enter code from email". Underneath is a "Code" label above a light gray input field containing the placeholder text "Enter Your Code Here". At the bottom are two buttons: a green "Login" button and a red "Close" button.

Рисунок 4.4.2.  
Окно авторизации  
(зарегистрированный  
пользователь)



A login form titled "Login" in purple. It features an "Email" label above a light gray input field containing the email "xibin29209@abudat.com". Below the input field is the text "You don't have account yet, choose nickname". Underneath is a "Nickname" label above a light gray input field containing the placeholder text "Enter Nickname". Below this is the text "Enter code from email". At the bottom is a "Code" label above a light gray input field containing the placeholder text "Enter Your Code Here". At the very bottom are two buttons: a green "Login" button and a red "Close" button.

Рисунок 4.4.3.  
Окно авторизации  
(новый пользователь)

## **Заключение**

В ходе выполнения курсовой работы был проведён анализ работы ленты коротких постов, 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 выше изложенные 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.