Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (МОСКОВСКИЙ ПОЛИТЕХ)

КУРСОВОЙ ПРОЕКТ

По курсу Разработка веб-приложений

ТЕМА

«Разработка веб-приложения для создания и управления опросами»

Выполнила: Бардина А.А.

Группа 231-329

Проверил: Кружалов А.С.

Москва, 2025

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc199591828)

[ВВЕДЕНИЕ 3](#_Toc199591829)

[РАЗДЕЛ 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc199591830)

[1.1 Обзор существующих программных продуктов по теме работы 4](#_Toc199591831)

[1.2. Анализ программных инструментов разработки веб-приложений 6](#_Toc199591832)

[1.3. Формулировка цели и задач работы 9](#_Toc199591833)

[РАЗДЕЛ 2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ 11](#_Toc199591834)

[2.1 Анализ целевой аудитории 11](#_Toc199591835)

[2.2 Описание функциональности приложения 12](#_Toc199591836)

[2.3 Проектирование модели данных 14](#_Toc199591837)

[2.4 Разработка макетов страниц 15](#_Toc199591838)

[РАЗДЕЛ 3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ 19](#_Toc199591839)

[3.1. Разработка базовой структуры приложения и вёрстка шаблонов страниц 19](#_Toc199591840)

[3.2. Реализация аутентификации пользователей 20](#_Toc199591841)

[3.3. Реализация CRUD-интерфейса для опросов 21](#_Toc199591842)

[3.4 Разработка механизма прохождения опросов 22](#_Toc199591843)

[3.5. Реализация системы аналитики 22](#_Toc199591844)

[РАЗДЕЛ 4. ОФОРМЛЕНИЕ ИТОГОВ РАБОТЫ 24](#_Toc199591845)

[4.1. Создание Git-репозитория с кодом проекта 24](#_Toc199591846)

[4.2. Деплой приложения на хостинг 24](#_Toc199591847)

[4.3. Оформление отчёта о проделанной работе 24](#_Toc199591848)

[ЗАКЛЮЧЕНИЕ 26](#_Toc199591849)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 27](#_Toc199591850)

ВВЕДЕНИЕ

Современные информационные технологии охватывают все больше аспектов нашей жизни, включая способы сбора, анализа и обработки данных. Одним из популярных инструментов для получения обратной связи и анализа мнений является система опросов, которая используется в различных сферах: от науки и бизнеса до образования и политики. В последние годы наблюдается активное развитие онлайн-опросников и сервисов для создания анкет, что значительно упрощает процесс сбора информации и позволяет проводить исследования на глобальном уровне.

Актуальность разработки веб-приложения для создания и управления опросами обусловлена несколькими факторами. Во-первых, в условиях растущей цифровизации и необходимости оперативного получения данных на различных уровнях организации, создание удобных и доступных платформ для организации опросов становится крайне важным. Во-вторых, на рынке существует множество решений, однако не все из них предлагают достаточно гибкие и удобные инструменты для пользователей, особенно в части настройки вопросов, анализа результатов и интеграции с другими сервисами.

Разработка веб-приложения для создания и управления опросами является важной и востребованной задачей, поскольку позволяет значительно ускорить процесс подготовки и проведения опросов, а также улучшить качество получаемых данных. Современные веб-технологии предоставляют широкие возможности для создания таких приложений, что позволяет эффективно решать задачи, связанные с динамичностью и масштабируемостью.

РАЗДЕЛ 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор существующих программных продуктов по теме работы

На текущий момент существуют различные программные продукты, предназначенных для создания и управления опросами. Эти решения предоставляют различные функции, такие как создание анкет, сбор и обработка данных, а также анализ полученных результатов. Рассмотрим наиболее популярные и широко используемые из них.

Google Forms

Google Forms — это бесплатный инструмент от Google, который позволяет создавать опросы и анкеты для сбора данных. Программа предоставляет пользователю простой и интуитивно понятный интерфейс, позволяя настраивать различные типы вопросов, а также интегрировать результаты с другими сервисами Google. Одним из преимуществ Google Forms является простота использования, а также возможность работы в реальном времени с несколькими пользователями. Однако, несмотря на свою доступность и функциональность, Google Forms имеет ограничения по кастомизации внешнего вида опросов и ограниченные аналитические возможности, также в Google Forms нельзя поделиться своим опросом в общем каталоге, либо просмотреть чужие опросы и также проголосовать в них.

SurveyMonkey

SurveyMonkey — это один из самых популярных коммерческих сервисов для создания и управления опросами. Он предоставляет широкий спектр функций, включая различные типы вопросов, возможность настройки дизайна опросов, а также инструменты для анализа результатов. В SurveyMonkey есть возможность создания сложных логик вопросов (например, на основе ответов пользователя показывать различные вопросы). Также сервис поддерживает интеграцию с различными внешними платформами и базами данных. Однако, для доступа ко всем функциональным возможностям требуется платная подписка, что может быть ограничивающим фактором для некоторых пользователей.

LimeSurvey

LimeSurvey — это открытое программное обеспечение для создания и управления онлайн-опросами. Оно предоставляет широкие возможности для настройки вопросов, логики переходов, а также позволяет проводить сложные многоуровневые исследования. LimeSurvey особенно популярен среди организаций, которые нуждаются в полной кастомизации своих опросов и хотят иметь полный контроль над данными. Однако, LimeSurvey требует определенных технических знаний для установки и настройки, что может быть неудобно для пользователей без опыта в разработке.

Qualtrics

Qualtrics — это мощная платформа для создания опросов, предназначенная в основном для корпоративного использования. Она предлагает широкие возможности для анализа данных и разработки сложных опросов с множеством условий и логик. Qualtrics используется крупными компаниями и исследовательскими организациями для проведения глубинных маркетинговых исследований и получения точных данных. Однако высокая стоимость подписки делает этот инструмент доступным только для крупных предприятий.

На основе анализа существующих программных продуктов можно выделить несколько ключевых особенностей и тенденций. Современные инструменты для создания опросов предлагают широкий функционал, включая разнообразие типов вопросов, гибкость настройки, а также аналитические возможности. Однако большинство доступных решений имеет свои ограничения, такие как недостаточная кастомизация внешнего вида опросов, ограниченные аналитические функции или высокая стоимость премиум-версий.

В рамках разработки собственного веб-приложения для создания и управления опросами важно учесть эти особенности и постараться предложить пользователю решение, которое будет сочетать удобство, функциональность и гибкость, удовлетворяя потребности как индивидуальных пользователей, так и организаций.

**1.2. Анализ программных инструментов разработки веб-приложений**

Разработка веб-приложения требует выбора подходящих инструментов, обеспечивающих удобство разработки, производительность, безопасность и масштабируемость системы. В современном веб-разработке используются различные языки программирования, фреймворки и базы данных. В данном разделе рассмотрены наиболее популярные технологии, применяемые в веб-разработке, с выделением тех, которые будут использоваться в данном проекте.

**Языки программирования**

Языки программирования играют ключевую роль в создании веб-приложений, разделяясь на инструменты для фронтенда (клиентская часть) и бэкенда (серверная часть).

**Фронтенд-языки:**

**HTML (HyperText Markup Language)** – язык разметки, используемый для создания структуры веб-страниц.

**CSS (Cascading Style Sheets)** – язык стилей, отвечающий за визуальное оформление элементов.

**JavaScript (JS)** – язык программирования, который позволяет делать веб-страницы интерактивными (динамическое изменение контента, анимации, обработка событий).

**Бэкенд-языки:**

**Python** – мощный высокоуровневый язык программирования с лаконичным синтаксисом и широкими возможностями, активно используемый в веб-разработке. В данном проекте будет использоваться Python с фреймворком Flask.

**PHP** – серверный язык программирования, традиционно применяемый для веб-разработки (например, с фреймворком Laravel).

**Java** – используется в крупных корпоративных проектах, особенно с фреймворком Spring Boot.

**JavaScript (Node.js)** – позволяет использовать один язык для фронтенда и бэкенда, благодаря среде выполнения Node.js.

**Фреймворки и библиотеки для фронтенда**

Фронтенд-фреймворки помогают ускорить процесс разработки интерфейса, обеспечивая готовые инструменты для работы с DOM, стилями и взаимодействием с сервером.

Популярные решения:

**React.js** – библиотека для построения компонентов интерфейса, широко применяемая в современных веб-приложениях.

**Vue.js** – легковесный фреймворк, который сочетает простоту и удобство в разработке.

**Angular** – мощный фреймворк от Google, подходящий для сложных веб-приложений.

В данной работе фронтенд будет реализован на **HTML, CSS и чистом JavaScript** без использования фреймворков. Однако, при необходимости, могут использоваться вспомогательные инструменты, такие как **Bootstrap** – CSS-фреймворк для удобной и быстрой стилизации интерфейса.

**Фреймворки для бэкенд-разработки**

Фреймворки для серверной части помогают организовать маршрутизацию запросов, работу с базой данных и безопасность приложения.

Популярные решения:

**Django (Python)** – мощный фреймворк для разработки веб-приложений с большим количеством встроенных инструментов.

**Flask (Python)** – легковесный микрофреймворк, который обеспечивает гибкость и минимализм.

**Express.js (Node.js)** – фреймворк для создания серверных приложений на JavaScript.

**Spring Boot (Java)** – мощное решение для корпоративных приложений.

**Laravel (PHP)** – удобный PHP-фреймворк с богатым функционалом.

В данном проекте выбран Flask, так как он позволяет быстро и удобно разрабатывать серверную часть.

**Системы управления базами данных (СУБД)**

СУБД играют ключевую роль в хранении данных и их обработке. Веб-приложения могут использовать реляционные или NoSQL-базы.

Реляционные базы данных (SQL):

**MySQL** – одна из самых популярных реляционных баз данных, отличающаяся высокой производительностью и надежностью. Она будет использоваться в данном проекте.

**PostgreSQL** – мощная реляционная СУБД с поддержкой сложных запросов.

**SQLite** – легковесная база данных, удобная для небольших приложений.

В данном проекте будет использоваться **MySQL**, так как она хорошо подходит для хранения структурированных данных, связанных с опросами и пользователями.

**Инструменты для разработки и развертывания**

Для удобной разработки и развертывания используются различные инструменты:

**Git и GitHub** – система контроля версий и платформа для хранения кода.

**Virtualenv** – инструмент для создания виртуального окружения Python, позволяющий изолировать зависимости проекта.

**Gunicorn** – WSGI-сервер, используемый для запуска Flask-приложений в продакшене.

В данном проекте будет использоваться **GitHub для контроля версий, Virtualenv для управления зависимостями и, при необходимости, Gunicorn для развертывания сервера**.

На основе анализа программных инструментов выбраны следующие технологии для разработки веб-приложения:

**Фронтенд:** HTML, CSS, JavaScript (возможно использование Bootstrap).

**Бэкенд:** Python с использованием Flask.

**База данных:** MySQL для хранения информации об опросах и пользователях.

**Развертывание:** GitHub, Virtualenv, Gunicorn.

Такой технологический стек обеспечит удобство разработки, высокую производительность и надежность веб-приложения.

1.3. Формулировка цели и задач работы

Целью данной курсовой работы является разработка веб-приложения для создания и управления опросами, обеспечивающего удобный интерфейс для пользователей, возможность гибкой настройки опросов и сбор аналитики на основе полученных ответов.

Для достижения поставленной цели необходимо решить следующие задачи:

**- Анализ существующих решений** – изучить существующие программные продукты для создания и управления опросами, выявить их преимущества и недостатки.

**- Выбор технологий разработки** – обосновать выбор инструментов для реализации веб-приложения, включая языки программирования, фреймворки и системы управления базами данных.

**- Проектирование архитектуры приложения** – определить структуру веб-приложения, разработать базу данных, продумать логику взаимодействия между клиентской и серверной частями.

**- Разработка пользовательского интерфейса** – создать удобный и интуитивно понятный интерфейс с использованием HTML, CSS и JavaScript.

**- Реализация серверной части** – разработать серверное приложение на Python с использованием Flask, обеспечив обработку запросов, управление пользователями и работу с базой данных.

**- Разработка функционала для работы с опросами** – реализовать возможность создания, редактирования, удаления и прохождения опросов, а также хранения и обработки ответов.

**- Интеграция с базой данных** – настроить взаимодействие между серверной частью и MySQL для хранения информации об опросах и пользователях.

**- Тестирование и отладка** – проверить работу веб-приложения, выявить и исправить возможные ошибки.

**- Развертывание приложения** – подготовить веб-приложение для работы на сервере и протестировать его в реальной среде.

**РАЗДЕЛ 2. ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ**

**2.1** Анализ целевой аудитории

Целевая аудитория веб-приложения охватывает широкий круг пользователей, различающихся по целям использования, уровню вовлечённости и интересу к содержанию опросов. Это как активные создатели контента, так и пассивные потребители информации — респонденты и наблюдатели.

**1. Создатели опросов**

Пользователи, создающие и публикующие опросы. Они используют приложение как инструмент для получения обратной связи, проведения исследований или просто обмена мнением.

**Их потребности включают:**

* Удобное создание и редактирование опросов
* Настройка видимости: публичный / по ссылке / приватный
* Получение и анализ результатов
* Возможность опубликовать опрос в общем каталоге

**2. Респонденты (участники опросов)**

Пользователи, проходящие опросы. Они могут перейти по ссылке или просматривать доступные опросы в каталоге.

**Им важны:**

* Простой и понятный интерфейс
* Конфиденциальность и анонимность ответов
* Доступ с любого устройства

**3. Наблюдатели / Исследователи / Интересующиеся**

Пользователи, не обязательно участвующие в опросах, но интересующиеся результатами. Они просматривают статистику, сравнивают мнения, анализируют ответы. Это могут быть:

* Студенты и преподаватели
* Маркетологи
* Люди, просто интересующиеся чужими точками зрения

**Им нужны:**

* Доступ к результатам публичных опросов
* Удобная визуализация статистики (графики, диаграммы)
* Возможность сортировки и фильтрации опросов по темам, дате, популярности

**Обобщённые характеристики целевой аудитории:**

* Разный уровень технической подготовки (интерфейс должен быть простым)
* Использование как ПК, так и мобильных устройств
* Интерес к взаимодействию: голосование, наблюдение, участие
* Ожидание прозрачности и удобства в получении информации

**2.2** Описание функциональности приложения

User story

**1. Как гость, я хочу просматривать статистику по опросам, чтобы понимать мнения других.**

**А значит, я хочу:**

* видеть список всех завершённых или активных опросов;
* открывать страницу с общей статистикой по каждому опросу;
* просматривать результаты в виде диаграмм;
* не проходить регистрацию и не логиниться;
* выгружать данные опросов в файл.

**2. Как авторизованный пользователь, я хочу проходить опросы, чтобы делиться мнением и видеть результаты.**

**А значит, я хочу:**

* зарегистрироваться и войти в систему;
* видеть список всех активных опросов;
* выбирать интересный опрос и ответить на его вопросы;
* отправлять ответы и быть уверенным, что они сохранены;
* один раз участвовать в каждом опросе;
* просматривать результаты после прохождения опроса;
* создавать новые опросы;
* добавлять к своим опросам вопросы и варианты ответов;
* иметь возможность удалять свои опросы;
* видеть статистику по своим опросам.

**4. Как администратор, я хочу управлять пользователями и контролировать опросы, чтобы поддерживать порядок в системе.**

**А значит, я хочу:**

* удалять пользователей;
* блокировать или удалять опросы, если они нарушают правила;
* видеть всю статистику по всем опросам, независимо от их автора;
* иметь полный доступ ко всем разделам приложения.

Use Case Diagram

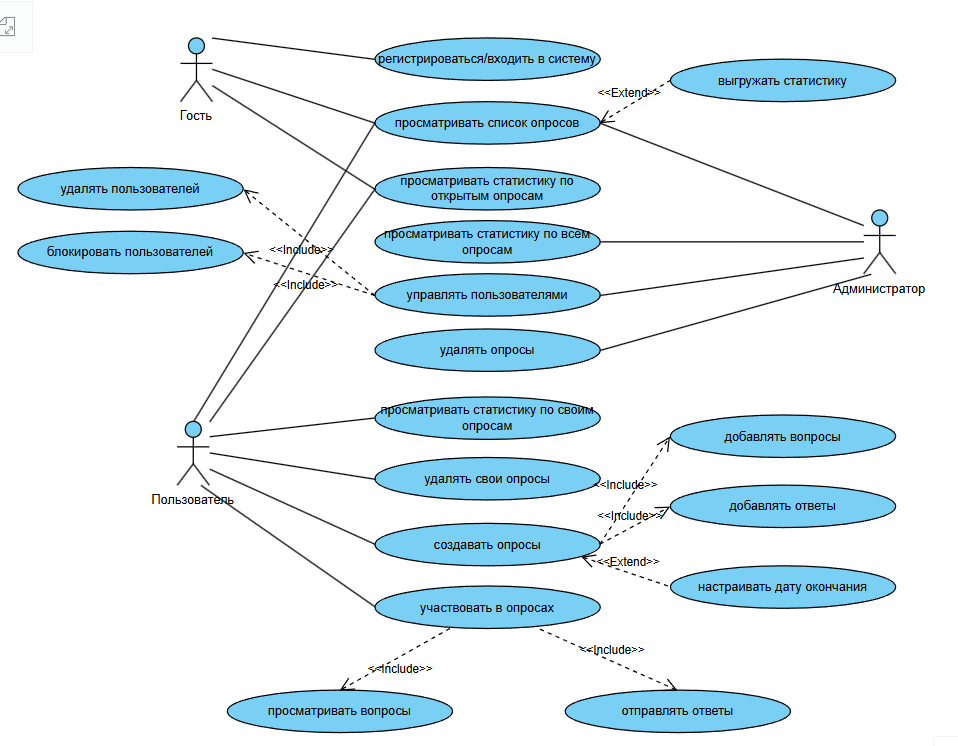


Рисунок - Use Case Diagram

2.3 Проектирование модели данных

ER-диаграмма

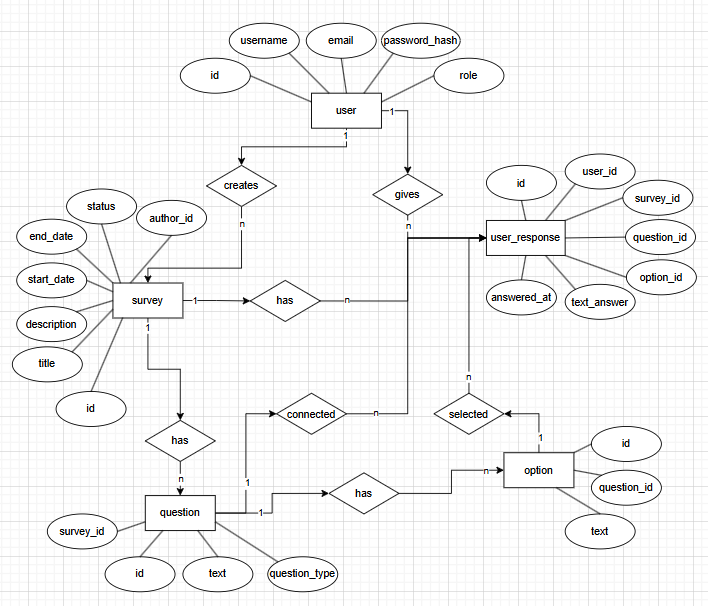


Рисунок - ER-диаграмма

Логическая схема базы данных

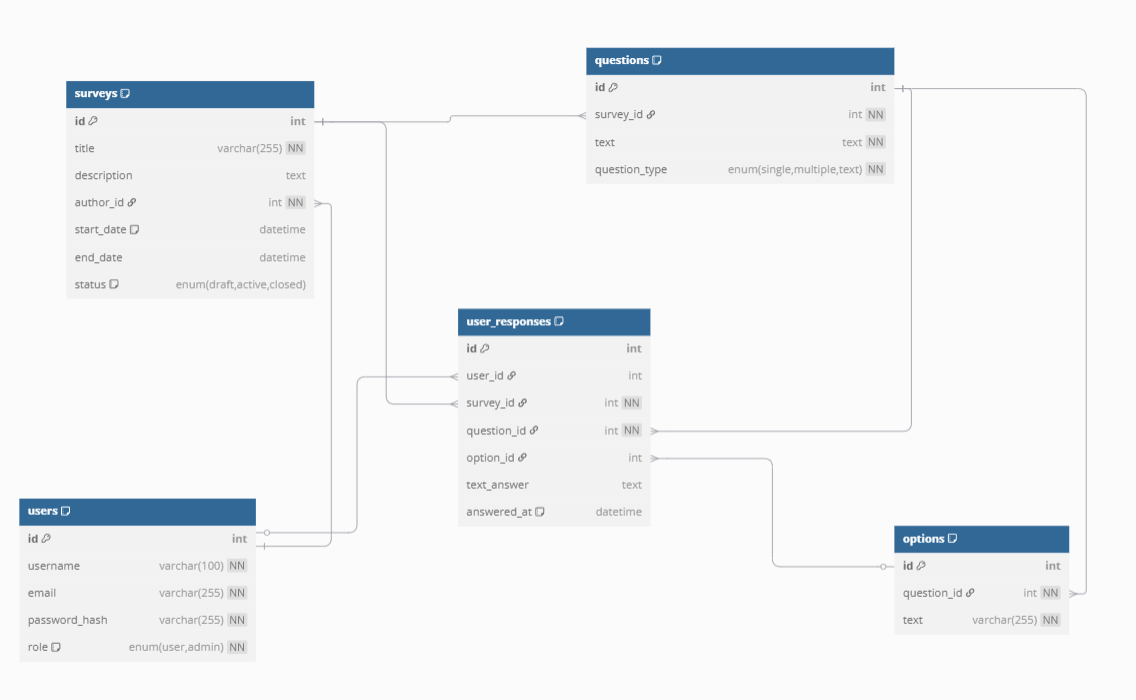


Рисунок - Логическая схема базы данных

Физическая схема базы данных для MySQL

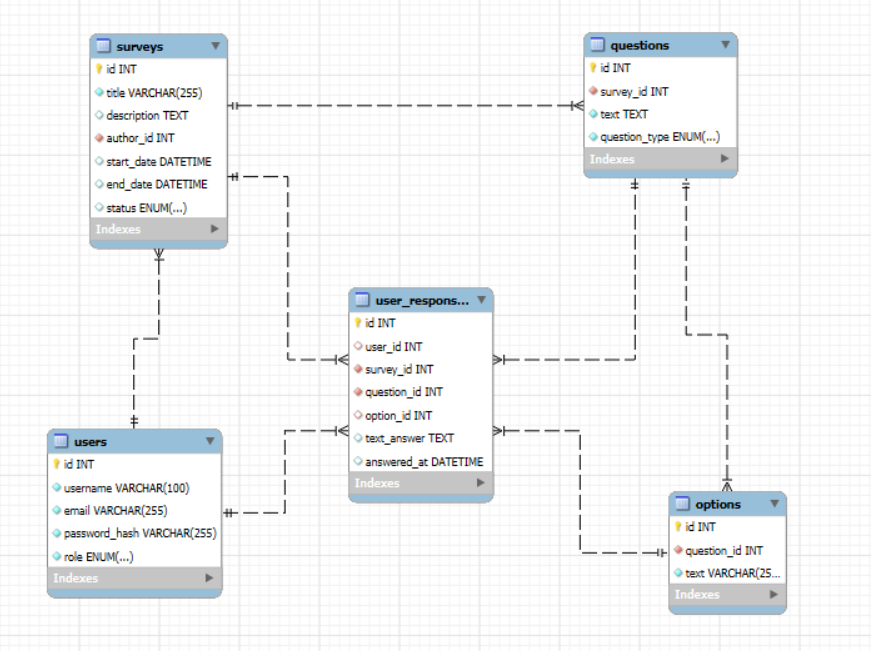


Рисунок - Физическая схема базы данных для MySQL

2.4 Разработка макетов страниц

Страницы входа и регистрации

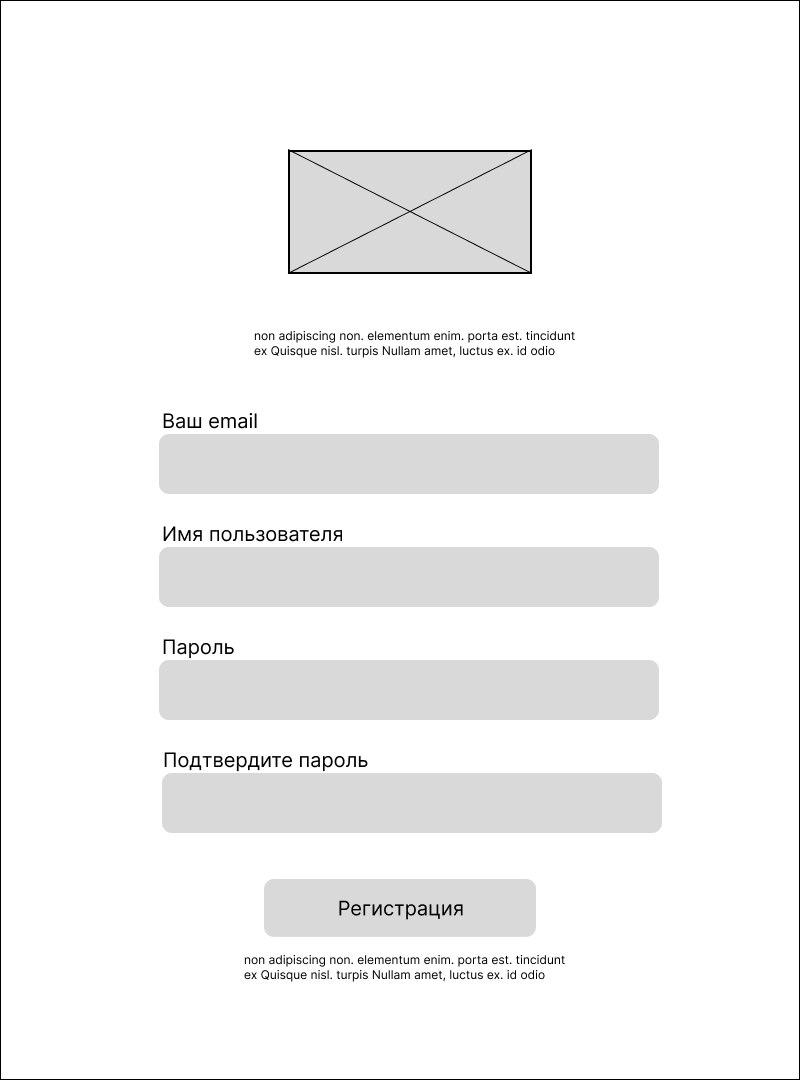


Рисунок - Страница регистрации

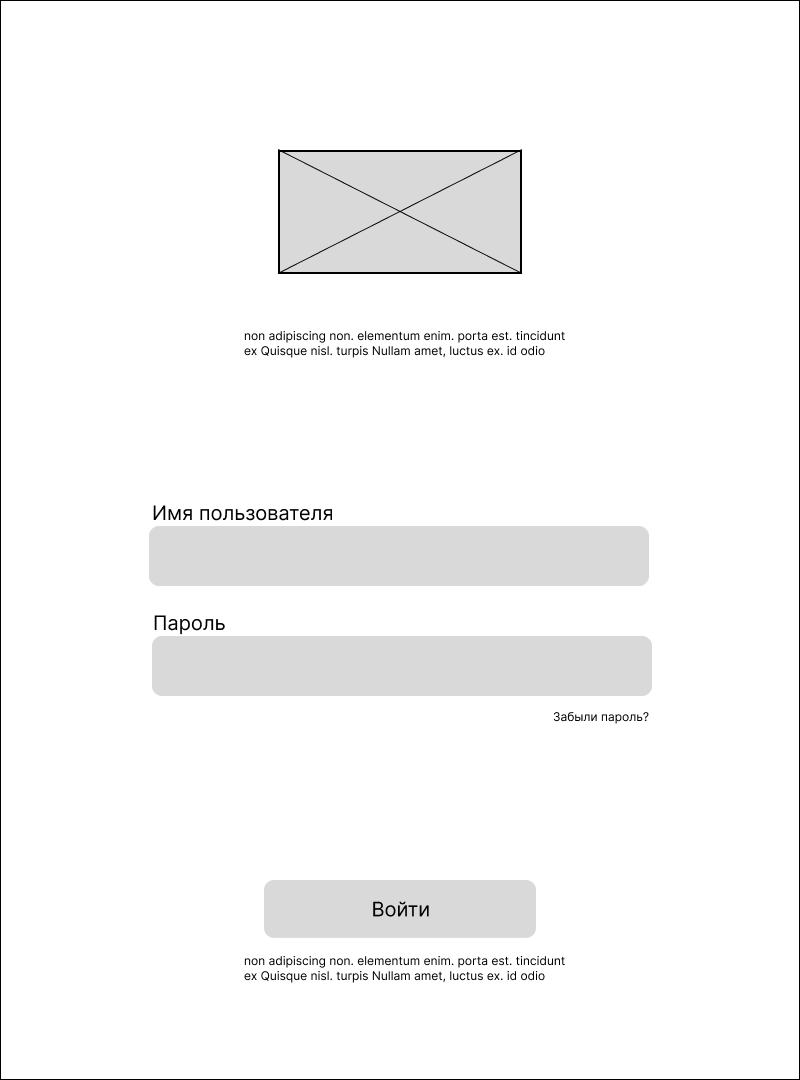


Рисунок - Страница входа

Страница общего каталога с опросами



Рисунок - Страница общего каталога с опросами

Страница создания опроса

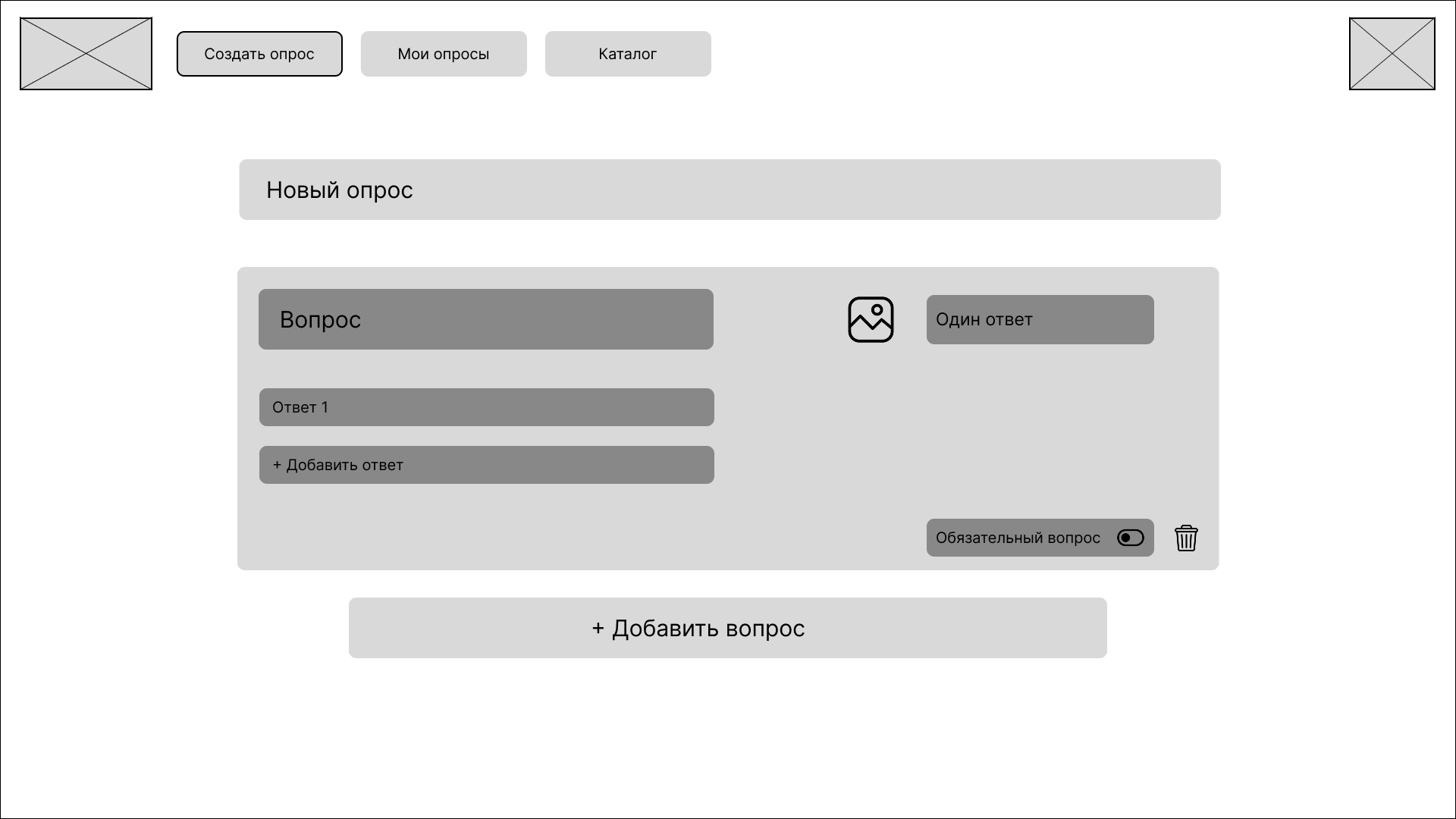


Рисунок - Страница создания опроса

Страница профиля

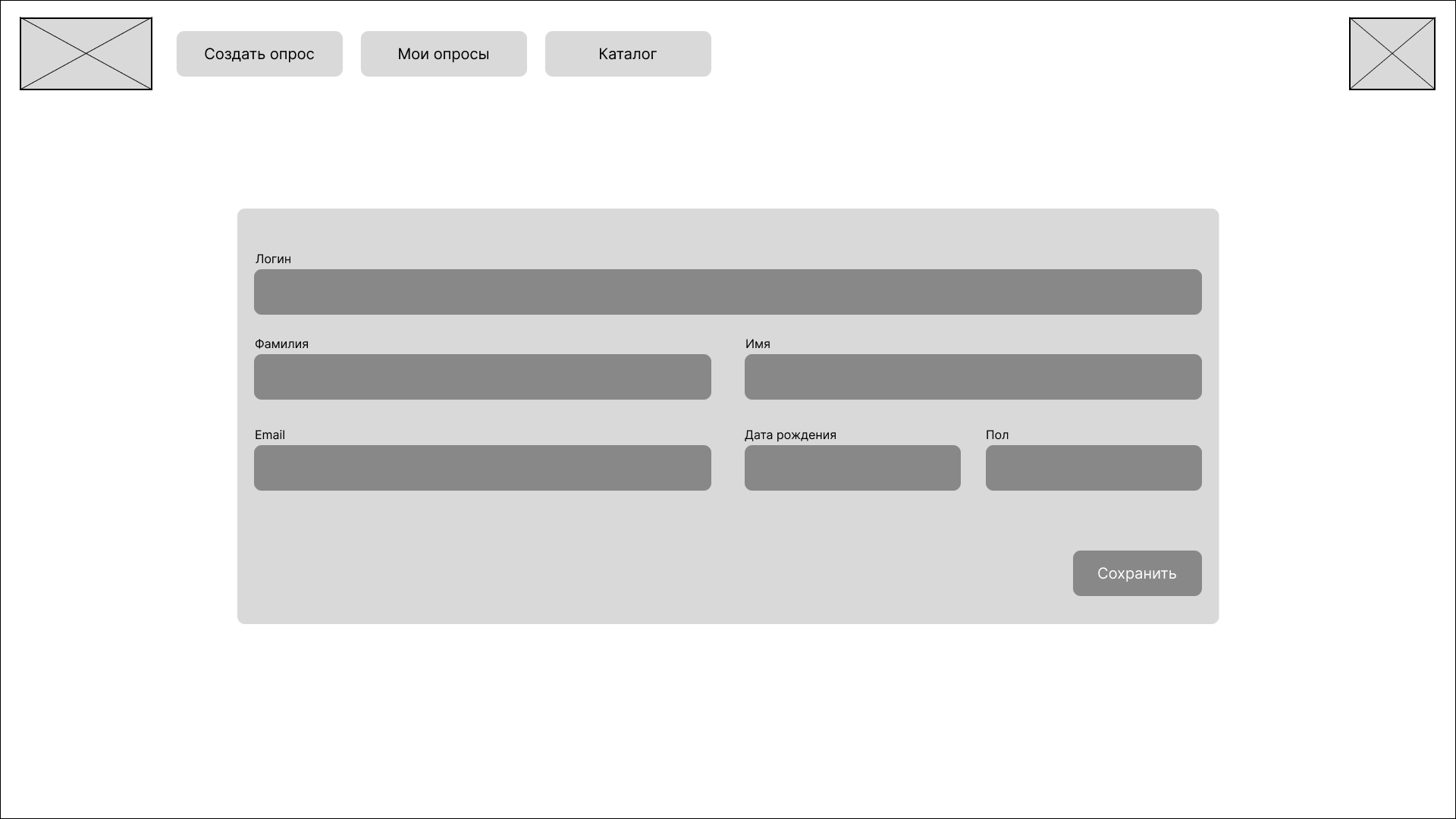


Рисунок - Страница профиля

Страница опросов пользователя

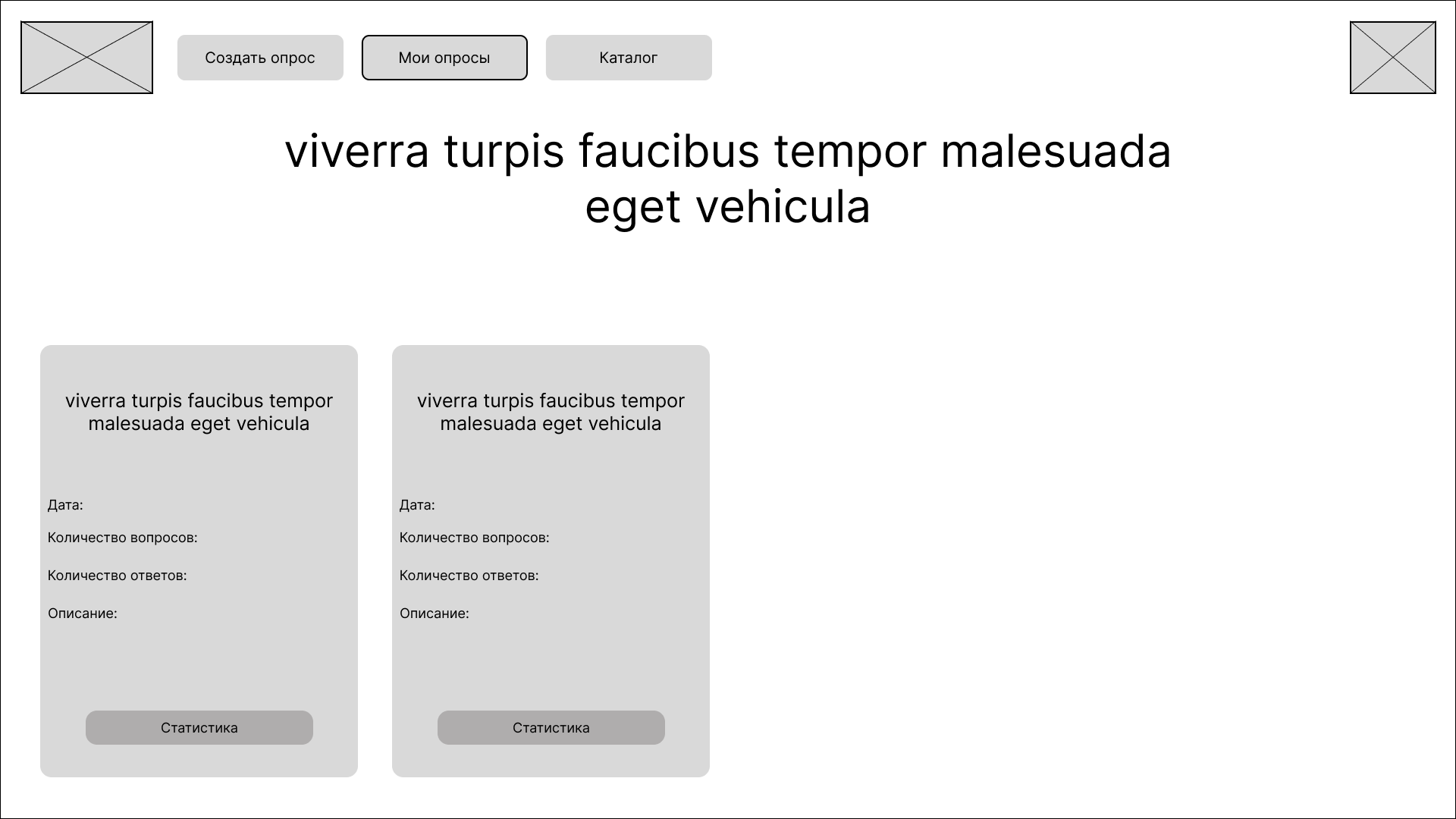


Рисунок - Страница опросов пользователя

Страница статистики конкретного опроса

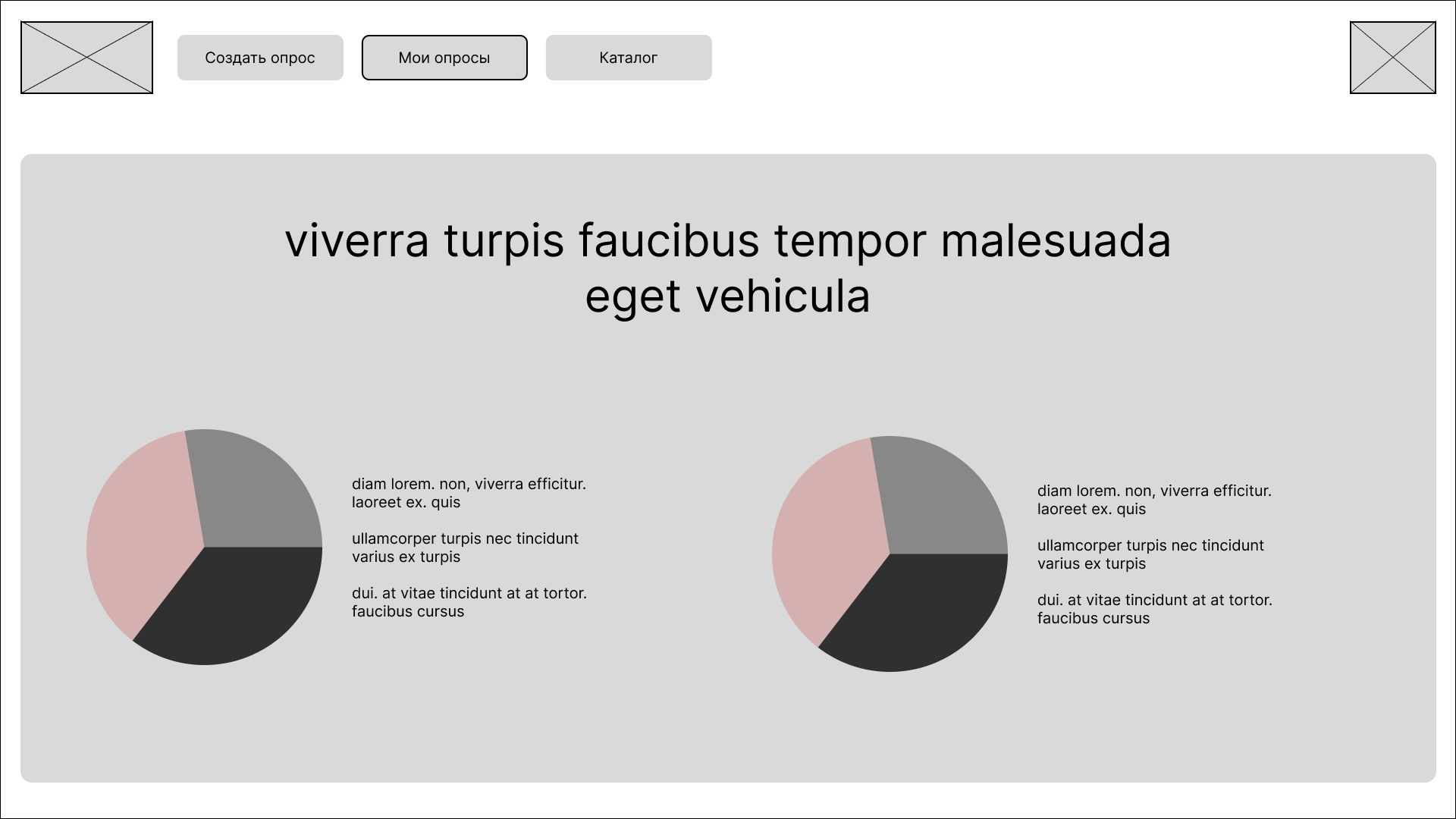


Рисунок - Страница статистики конкретного опроса

РАЗДЕЛ 3. РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ

Разработка веб-приложения велась с использованием микрофреймворка Flask, который обеспечивает гибкость, понятную архитектуру и простоту развертывания. В ходе реализации были использованы такие компоненты как Flask Blueprints для маршрутизации, SQLAlchemy для работы с базой данных и Jinja2 для шаблонов.

3.1. Разработка базовой структуры приложения и вёрстка шаблонов страниц

Приложение имеет модульную структуру. Основной код разделён на следующие компоненты:

routes/ — маршруты, сгруппированные по функциональности;

repositories/ — репозитории для работы с базой данных;

templates/ — шаблоны HTML с использованием Jinja2;

static/ — стили, скрипты и изображения;

utils/ — вспомогательные функции;

models.py — ORM-модели.

Вёрстка реализована с помощью шаблонов Jinja2. Базовый шаблон base.html содержит общую структуру и подключает стили:

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>{% block title %}Опросы{% endblock %}</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

</head>

<body>

{% block content %}{% endblock %}

</body>

</html>

Другие шаблоны (например, login.html, catalog.html, results.html) наследуют base.html:

{% extends 'base.html' %}

3.2. Реализация аутентификации пользователей

Для регистрации и входа пользователей реализован blueprint auth\_routes.py. Аутентификация выполнена с использованием библиотеки Flask-Login.

Фрагмент маршрута авторизации:

@bp.route('/login', methods=['GET', 'POST'])

def login():

if request.method == 'POST':

username = request.form.get('username')

password = request.form.get('password')

remember\_me = request.form.get('remember\_me', None) == 'on'

user = user\_repository.get\_user\_by\_username\_and\_password(username, password)

if user:

flash('Авторизаия прошла успешно', 'success')

login\_user(user, remember=remember\_me)

return redirect(url\_for('survey.catalog'))

flash('Неверное имя пользователя или пароль', 'danger')

return render\_template('auth/login.html')

Функция user\_repository.get\_user\_by\_username\_and\_password реализована в модуле user\_repository.py:

class UserRepository:

def get\_user\_by\_username\_and\_password(self, username, password):

user = User.query.filter\_by(username=username).first()

if user and user.check\_password(password):

return user

return None

Использование репозиториев позволяет изолировать бизнес-логику от запросов к базе данных.

3.3. Реализация CRUD-интерфейса для опросов

Каждый авторизованный пользователь может создавать, удалять и просматривать опросы. Эти функции реализованы в survey\_routes.py с использованием SurveyRepository.

Пример создания опроса:

@bp.route('/create', methods=['GET', 'POST'])

@login\_required

def create\_survey():

if request.method == 'POST':

title = request.form['title']

description = request.form['description']

survey\_repository.create(title, description, current\_user.id)

flash('Опрос создан!', 'success')

return redirect(url\_for('main.catalog'))

return render\_template('create\_survey.html')

Функция create в репозитории:

class SurveyRepository:

def create(self, title, description, user\_id):

new\_survey = Survey(title=title, description=description, author\_id=user\_id)

db.session.add(new\_survey)

db.session.commit()

3.4 Разработка механизма прохождения опросов

Пользователи могут проходить опубликованные опросы, отвечая на вопросы. Все ответы сохраняются в базу данных. Реализация механизма включает отображение вопросов и сохранение результатов.

Пример маршрута:

@bp.route('/<int:survey\_id>/take', methods=['GET', 'POST'])

def take\_survey(survey\_id):

survey = survey\_repository.get\_survey\_by\_id(survey\_id=survey\_id)

return render\_template('survey/take.html', survey=survey)

3.5. Реализация системы аналитики

Раздел аналитики позволяет просматривать результаты опросов: количество участников, распределение ответов. Данные извлекаются из базы через агрегатные SQL-запросы в survey\_repository.py.

Эти данные передаются в шаблон и отображаются в виде различных графиков (с помощью Chart.js).

3.6. Разработка механизма экспорта данных

Была реализована возможность экспорта результатов опросов в формате CSV. Для этого используется стандартный модуль csv. Маршрут описан в stats.py и отображается с помощью шаблона survey\_stats.html

На данном этапе реализован полноценный backend и frontend с поддержкой аутентификации, CRUD-интерфейсом для опросов, прохождением опросов, аналитикой и экспортом данных. Использование репозиториев, шаблонов и модульной структуры сделало код удобным для сопровождения и масштабирования.

РАЗДЕЛ 4. ОФОРМЛЕНИЕ ИТОГОВ РАБОТЫ

Завершающий этап разработки включал подготовку итоговых материалов: публикацию проекта в системе контроля версий, развёртывание веб-приложения на сервере и оформление отчётной документации. Эти задачи направлены на обеспечение доступности, повторяемости и демонстрации результатов проделанной работы.

4.1. Создание Git-репозитория с кодом проекта

Для хранения и управления версионированием исходного кода проекта был использован Git. Репозиторий был создан на платформе GitHub.

На протяжении всей разработки использовались следующие принципы:

* регулярные коммиты с осмысленными сообщениями;
* использование .gitignore для исключения конфиденциальных и временных файлов (например, .env, \_\_pycache\_\_, migrations/versions и т.д.);
* документирование зависимостей в файле requirements.txt с помощью pip freeze.

Такой подход обеспечил прозрачность истории разработки и упростил работу, отладку и развёртывание проекта.

4.2. Деплой приложения на хостинг

Для публикации веб-приложения был использован бесплатный хостинг Pythonanywhere. Развёртывание производилось по следующему плану:

* Указание переменных окружения (.env) через панель управления хостинга.
* Настройка базы данных MySQL.
* Деплой в основную ветку репозитория с помощью Git.

4.3. Оформление отчёта о проделанной работе

В завершение была подготовлена курсовая работа, отражающая:

* архитектуру и структуру проекта;
* этапы реализации;
* схемы, примеры кода и интерфейсы приложения;
* описание развёртывания и работы с репозиторием.

ЗАКЛЮЧЕНИЕ

В рамках курсовой работы было спроектировано и реализовано полноценное веб-приложение для создания, прохождения и анализа опросов. В качестве платформы выбран фреймворк Flask, что позволило обеспечить модульную структуру и простоту вёрстки с использованием шаблонов Jinja2.

Основные функции приложения включают:

* аутентификацию пользователей;
* CRUD-интерфейс для опросов;
* возможность прохождения опросов и сохранения ответов;
* просмотр результатов и базовую аналитику;
* экспорт данных в CSV.

Особое внимание уделялось чистоте архитектуры: использованы Blueprint’ы, паттерн "репозиторий", единый стиль написания кода. Приложение успешно развёрнуто на хостинге и доступно для использования.

Результаты курсового проекта могут быть использованы в дальнейшем как основа для более сложной системы: с расширенной аналитикой, интеграцией внешних API и т.д.

Таким образом, поставленные задачи были решены в полном объёме, а полученный результат соответствует целям проекта.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Документация Flask [Электронный ресурс]. — Режим доступа: <https://flask.palletsprojects.com/>
2. Документация SQLAlchemy [Электронный ресурс]. — Режим доступа: <https://docs.sqlalchemy.org/>
3. Документация Jinja2 [Электронный ресурс]. — Режим доступа: <https://jinja.palletsprojects.com/>
4. Документация Bootstrap [Электронный ресурс]. — Режим доступа: <https://getbootstrap.com/>
5. Документация GitHub [Электронный ресурс]. — Режим доступа: <https://docs.github.com/>
6. Render Docs: Развёртывание Flask-приложений [Электронный ресурс]. — Режим доступа: <https://render.com/docs/deploy-flask>
7. PythonAnywhere Help: Развёртывание Flask-приложения [Электронный ресурс]. — Режим доступа: <https://help.pythonanywhere.com/pages/Flask/>