РЕAЛИЗАЦИЯ TO-DO LIST

**Детальный документ с требованиями для проекта** Nick

**1. Обзор проекта**

**Название проекта**: Nick

**Описание**: Nick — это веб-приложение для геймифицированных прогулок и обучения по уходу за собаками и окружающей средой, а также геймифицированных интерактивных экскурсий по историческим местам и достопримечательностям города Санкт-Петербург. Приложение использует нейросети для создания уникальных задач и квестов, которые помогают владельцам собак и их питомцам весело проводить время на свежем воздухе, а также учиться заботиться о природе, а еще нужны для . создания уникальных задач и квестов, которые помогут пользователям не только узнать больше о месте, но и сделать процесс познавательным и увлекательным

**2. Основные функции**

**2.1 Геймифицированные задачи**

* **Описание**: Пользователи получают задания, которые можно выполнить во время прогулок с собакой.
* **Примеры задач**:
  + Найти определенные растения.
  + Убрать мусор в парке.
  + Выполнить упражнения с питомцем.
  + Загадки и ребусы, требующие перемещения по окрестностям.

**2.2 Интерактивные карты**

* **Описание**: Приложение предоставляет интерактивные карты с указанием ближайших парков, мест для выгула собак и экопунктов.
* **Функции**:
  + Показ ближайших точек интереса.
  + Открытие новой точки на карте после выполнения задачи.

**2.3 Обратная связь и образовательный контент**

* **Описание**: Приложение предоставляет краткую информацию о растениях, животных и экологических проблемах, встречающихся в данном районе.
* **Форматы контента**:
  + Сторис с короткими видео.
  + Анимации.
  + Тексты.

**2.4 Социальные элементы**

* **Описание**: Пользователи могут делиться своими достижениями в соцсетях и соревноваться с друзьями.
* **Функции**:
  + Деление достижениями.
  + Соревнования между пользователями.

**2.5 Персонализация**

* **Описание**: На основе предпочтений пользователя и его предыдущих действий нейросеть подбирает наиболее интересные и подходящие задачи и маршруты.
* **Функции**:
  + Персонализированные задачи и маршруты.
  + Рекомендации новых мест для прогулок.

**3. Технические детали**

**3.1 Архитектура системы**

* **Бэкенд**:
  + Фреймворк: Django/Flask
  + Язык: Python
  + База данных: PostgreSQL/MySQL
* **Фронтенд**:
  + Фреймворк: React/Vue/Angular
  + Язык: JavaScript/TypeScript
* **Нейросети**: TensorFlow/PyTorch
* **API**: RESTful, JWT для аутентификации
* **Картографический сервис**: Google Maps API

**3.2 Сбор данных**

* **Источник данных**:
  + Данные о растениях, животных и экологических проблемах.
  + Пользовательские данные (предпочтения, выполненные задачи и т.д.).

**3.3 Безопасность**

* **Аутентификация**: JWT
* **Хранение данных**: Шифрование пользовательских данных.
* **Защита от атак**: Использование best practices для защиты от XSS, CSRF и SQL Injection.

**4. Пользовательские сценарии**

**4.1 Регистрация и вход в систему**

* **Пользовательский сценарий**:
  1. Пользователь заходит на сайт.
  2. Пользователь регистрируется, вводя email, пароль и другие данные.
  3. Пользователь подтверждает email.
  4. Пользователь входит в систему, вводя email и пароль.

**4.2 Выполнение задач**

* **Пользовательский сценарий**:
  1. Пользователь выбирает задачу из списка доступных.
  2. Пользователь выполняет задачу на улице (например, найти растение).
  3. Пользователь отмечает выполнение задачи в приложении.
  4. Пользователь получает образовательный контент и переходит к следующей задаче.

**4.3 Использование карты**

* **Пользовательский сценарий**:
  1. Пользователь открывает карту.
  2. Пользователь видит ближайшие парки, места для выгула собак и экопункты.
  3. Пользователь выбирает точку на карте и получает маршрут до неё.

**4.4 Социальное взаимодействие**

* **Пользовательский сценарий**:
  1. Пользователь выполняет задачу и делится достижением в соцсетях.
  2. Пользователь видит достижения друзей и соревнуется с ними.

**4.5 Персонализация задач**

* **Пользовательский сценарий**:
  1. Пользователь выполняет задачи и взаимодействует с приложением.
  2. Нейросеть анализирует действия пользователя.
  3. Приложение предлагает персонализированные задачи и маршруты.

**5. План разработки**

**5.1 Этап 1: Подготовка**

* Создание репозитория на GitHub.
* Настройка виртуального окружения и установка зависимостей.
* Определение структуры базы данных.

**5.2 Этап 2: Разработка бэкенда**

* Реализация API для аутентификации пользователей.
* Создание моделей и представлений для задач и контента.
* Интеграция с Google Maps API.

**5.3 Этап 3: Разработка фронтенда**

* Разработка основных компонентов интерфейса.
* Интеграция с бэкендом через API.
* Реализация интерактивной карты.

**5.4 Этап 4: Интеграция нейросети**

* Сбор и подготовка данных для обучения.
* Обучение моделей нейросети.
* Интеграция моделей в бэкенд.

**5.5 Этап 5: Тестирование и развертывание**

* Юнит-тестирование компонентов.
* Интеграционное тестирование системы.
* Развертывание на сервере и публикация в сети.

**5.6 Этап 6: Пострелизная поддержка**

* Сбор обратной связи от пользователей.
* Регулярные обновления и исправление ошибок.