**ЛИЦЕЙ АКАДЕМИИ ЯНДЕКСА**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

на разработку проекта по блоку «Web + Flask»  
на тему  
**«Интерактивный сайт для онлайн-раскладов Таро»**

**1. Общие сведения**

**1.1 Полное название приложения**

**Интерактивный веб-сервис онлайн-раскладов Таро**

**1.2 Сокращённое название приложения**

**Таро Онлайн**

**1.3 Назначение приложения**

Приложение предназначено для пользователей, желающих получить быстрый доступ к случайным раскладам Таро по выбранному типу ("карта дня", "да/нет", "совет"). Сервис выполняет функцию генерации раскладов и визуализации карты с описанием. Может быть расширен под административный интерфейс в будущем.

**1.4 Разработчики**

Проект разработан Стариковой Варварой и Щербаковой Татьяной ученицами Лицея Академии Яндекса на базе центра цифрового образования детей «IT-Куб».

**2. Требования к средствам реализации приложения**

**2.1 Технологический стек**

* Язык программирования: **Python 3.12**
* Web-фреймворк: **Flask**
* СУБД: **SQLite3**
* HTML, CSS, JavaScript (Vanilla)
* CSS-фреймворк: **Bootstrap 5**

**2.2 Инструментальные средства**

* Среда разработки: **VS Code / PyCharm**
* Утилиты: **sqlite3 CLI** или **DB Browser for SQLite**
* Git (опционально, для контроля версий)
* Браузер: Google Chrome, Firefox и др.

**2.3 Описание требований к платформе**

* Операционная система: Windows / macOS
* Поддержка Python ≥ 3.10
* Интернет-соединение не требуется (работает локально)
* Для серверной версии: поддержка WSGI и Python
* Клиентские рабочие места: браузер с поддержкой JavaScript и HTML5

**3. Требования к функциональной составляющей**

**3.1 Общий состав функций**

* Отображение главной страницы с кнопками выбора типа расклада
* Обработка кнопок: "карта дня", "да/нет", "совет"
* По нажатию на кнопку "разложить":
  + Получение случайной карты из базы по выбранному типу
  + Отображение описания карты
  + Отображение изображения карты (если указано в БД)
* Реализация взаимодействия с сервером без перезагрузки страницы (AJAX)

**3.2 Состав данных, подлежащих хранению**

В базе данных cards.db хранится таблица cards, содержащая:

* id: уникальный идентификатор
* name: название карты
* description: описание карты
* spread: тип расклада (например: "карта дня", "совет")
* photo: имя изображения карты (файл в папке /static/img/)

СУБД: **SQLite3**

**3.3 Требования к интерфейсу**

* Адаптивный и минималистичный интерфейс на Bootstrap
* Основные компоненты:
  + Кнопки выбора расклада (3 штуки)
  + Кнопка "разложить"
  + Блок заголовка (отображает название выбранного расклада)
  + Блок с описанием карты
  + Блок изображения карты
* Интерфейс не должен перезагружать страницу при действиях пользователя
* Неправильные действия (например, "разложить" без выбора) обрабатываются через alert()

**3.4 Требования к обработке данных**

* При выборе типа расклада, значение записывается в JS-переменную
* При нажатии "разложить", фронтенд отправляет GET-запрос на /get-card?spread=...
* Сервер выбирает случайную запись из таблицы по фильтру spread = значение
* Результат передаётся в JSON-формате
* JS обновляет:
  + текстовое поле (описание карты)
  + изображение (если указано)
* В случае отсутствия карт с заданным раскладом сервер возвращает ошибку