**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**  
**«** **Национальный исследовательский технологический университет**

**«МИСИС»**

**Институт компьютерных наук**

**Кафедра инженерной кибернетики**

**Прикладная математика**

КУРСОВАЯ РАБОТА

**по дисциплине «Объектно-ориентированное программирование»**

**Веб-приложение «Карточки для изучения английских слов»**

Выполнили:

студенты 1 курса группы БПМ-24-2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Кузьменко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. И. Джекиева

Преподаватель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. В. Еременко

**Москва 2025**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc1588991755)

[1 ОПИСАНИЕ ЗАДАЧИ 4](#_Toc300184645)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 5](#_Toc1484180289)

[2.1 Общие требования 5](#_Toc955169772)

[2.2 Главная страница 5](#_Toc1320953720)

[2.3 Страница изучения слов 5](#_Toc2127411033)

[2.4 Алгоритм интервального повторения 6](#_Toc446393116)

[2.5 API и взаимодействие с сервером 7](#_Toc1288705436)

[3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 8](#_Toc1729925595)

[3.1 Установка и запуск 8](#_Toc1576064731)

[3.2 Работа с приложением 8](#_Toc1594776005)

[3.3 Особенности и советы 9](#_Toc1750874149)

[4 ТЕХНИЧЕСКОЕ ОПИСАНИЕ ДЕТАЛЕЙ РЕАЛИЗАЦИИ 11](#_Toc286506280)

[4.1 Общая архитектура 11](#_Toc1734361166)

[4.2 Структура проекта 11](#_Toc23858830)

[4.3 Формат и содержание cards.json 11](#_Toc315086510)

[4.4 Тестирование 12](#_Toc21389315)

[5 СБОРКА И РАЗВЁРТЫВАНИЕ 13](#_Toc1117826283)

[5.1 Требования к ПО 13](#_Toc294894742)

[5.2 Сборка и запуск 13](#_Toc1633953531)

[6 ПРИЛОЖЕНИЯ 14](#_Toc486558662)

[Приложение А. Скриншоты интерфейса 14](#_Toc425425950)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc1626488858)

### ВВЕДЕНИЕ

В рамках данной курсовой работы применяются технологии программирования при разработке веб-приложения «Карточки для изучения английских слов». Веб-приложение предназначено для пользователей, которые хотят эффективно запомнить английские слова начинающего уровня.

### 1 ОПИСАНИЕ ЗАДАЧИ

Разработать веб-приложение «Карточки для изучения английских слов» с использованием метода интервального повторения (Spaced Repetition).

Для приложения разрабатывается минималистичный дизайн, интуитивно понятный пользователю. Приложение помогает эффективно запоминать слова за счёт алгоритма интервального повторения. Доступен просмотр слов по разным темам. Прогресс пользователя сохраняется, есть возможность сброса прогресса.

### 2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

##### 2.1 Общие требования

Разработать веб-приложение «Карточки для изучения английских слов» на языке программирования C++ с использованием принципов объектно-ориентированного программирования.

##### 2.2 Главная страница

Главная страница демонстрирует название («English Cards») и приветствует пользователя. Далее пользователю доступна кнопка «Начать» для перехода на вторую страницу изучения слов.

##### 2.3 Страница изучения слов

На странице обучения (cards.html) реализован следующий интерфейс и сценарии взаимодействия:

1. **Выбор темы**

В верхней части страницы отображается горизонтальный список тем — каждая тема представлена кнопкой-табом. Всего доступно семь тематических подборок. Активная кнопка подсвечивается более насыщенным цветом, что позволяет пользователю мгновенно увидеть, какая тема выбрана и что именно он сейчас изучает.

1. **Основная карточка**

Посередине экрана находится крупная карточка, на которой отображается одно слово на английском языке. Справа от слова расположен значок динамика, по нажатию на который запускается Web Speech API — слово проговаривается вслух (при условии поддержки браузером). Такая озвучка помогает не только визуальному, но и аудиальному запоминанию лексики.

1. **Информация о повторении**

Под словом выводятся два важных параметра:

* 1. **Текущий интервал** (в минутах или часах), показывающий, через сколько времени карточка будет показана вновь, если ответ «Помню».
  2. **Время до следующего повторения**, отображаемое в формате «X мин» или «Y ч». Если карточка новая, параметр показывает «Сейчас», что означает, что карточку можно сразу изучать.

Доступные интервалы заданы в порядке возрастания: 0 мин (новая карточка), 1 мин, 5 мин, 30 мин, 1440 мин (24 ч). При достижении максимального интервала дальнейшее его увеличение не происходит — карточка будет показываться каждые 24 часа.

1. **Показ перевода и ответы**

В нижней части карточки располагается кнопка **«Показать перевод»**. При её нажатии под английским словом появляется его русский эквивалент, оформленный заметным шрифтом или цветом. После этого появляется блок с двумя большими кнопками:

* 1. **«Не помню»** — при выборе этого ответа интервал повторения сбрасывается до 0, и карточка сразу же возвращается в очередь для повторения.
  2. **«Помню»** — интервал повторения повышается на один шаг, а карточка временно убирается из текущей очереди до момента следующего показа.

1. **Прогресс-бар**

Над кнопками расположен визуальный индикатор прогресса: горизонтальная полоска, заполненная на X% в соответствии с количеством уже изученных из общей массы карточек темы. Справа от полоски выводится текстовый счётчик «X/Y», где X — число пройденных карточек, Y — общее число карточек в теме.

1. **Конец темы**

Когда в выбранной теме не остаётся карточек, подходящих для повторения (либо все изучены, либо ждут своего времени), на месте карточки выводится сообщение **«Нет карточек для повторения»** и подсказка: «Выберите другую тему или вернитесь позже». Кнопка «Показать перевод» и блок ответов при этом скрываются, а прогресс-бар остаётся пустым.

##### 2.4 Алгоритм интервального повторения

Интервальное повторение построено по классической пятишаговой схеме:

1. **Новая карточка**: интервал = 0 мин → карточка показывается сразу до первого ответа.
2. **Первый правильный ответ** («Помню»): интервал увеличивается до 1 мин.
3. **Второй правильный ответ**: интервал = 5 мин.
4. **Третий правильный ответ**: интервал = 30 мин.
5. **Четвёртый и последующие правильные ответы**: интервал = 1440 мин (24 ч).

Если пользователь выбирает «Не помню» на любом шаге, интервал сбрасывается до 0 и карточка снова становится новой. После каждого ответа рассчитывается время следующего показа:

nextReview = текущая отметка времени + interval (в минутах)

Таким образом, система гибко адаптируется к уровню знаний и скорости восприятия каждого пользователя.

##### 2.5 API и взаимодействие с сервером

Серверная часть реализована на C++ и отвечает за хранение и обновление справочного файла cards.json, содержащего темы, слова и их прогресс. Доступны следующие REST-эндпоинты:

1. **GET /api/topics**

Возвращает JSON-массив строк, каждая из которых — название доступной темы.

1. **GET /api/cards**

Возвращает полный JSON-объект с темами и массивами карточек по каждой теме.

1. **GET /api/cards/{topic}**

Возвращает массив карточек для заданной темы {topic}. Если тема не найдена, возвращается объект с ошибкой.

1. **GET /api/cards/{topic}/{id}/progress**

Возвращает JSON-объект прогресса карточки с идентификатором {id} в теме {topic} (поля interval и nextReview).

1. **PUT /api/cards/{topic}/{id}**

Принимает в теле JSON вида



и обновляет поля соответствующей карточки в файле. В ответ возвращает { "success": true } или { "error": "..." }.

Все ответы сервера имеют заголовок Content-Type: application/json и поддерживают CORS (заголовок Access-Control-Allow-Origin: \*), что позволяет клиенту беспрепятственно обращаться к API.

### 3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

##### 3.1 Установка и запуск

1. **Сервер**
   1. Скачайте исходный код серверной части.
   2. Открыть build.bat или build.sh для MacOS (для сборки и запуска сервера).

На экране появится сообщение Server started at <http://localhost:8080>.

1. **Клиент**
   1. Откройте папку frontend
   2. Запустите файл index.html в браузере.

##### 3.2 Работа с приложением

1. **Начало обучения**
   1. На главной странице нажмите **«Начать»**. Вы попадёте на страницу изучения слов.
2. **Выбор темы**
   1. В верхней панели нажмите на название интересующей темы. Активная тема подсветится, и приложение загрузит все доступные для повторения карточки.
3. **Изучение карточек**
   1. **Чтение слова**: слово отображается крупным шрифтом.
   2. **Озвучка**: нажмите на значок динамика справа от слова, чтобы услышать правильное произношение.
   3. **Показ перевода**: нажмите **«Показать перевод»** — появится перевод под английским словом.
   4. **Выбор ответа**: нажмите либо **«Не помню»**, либо **«Помню»**.
      1. При **«Не помню»** карточка остаётся в начале очереди, интервал сбрасывается до 0.
      2. При **«Помню»** карточка уходит на время, определяемое новым интервалом (1, 5, 30 мин или 24 ч).
   5. **Прогресс**: прогресс-бар и счётчик «X/Y» обновляются после каждого ответа.
4. **Переход к следующему слову**
   1. После выбора ответа приложение автоматически показывает следующую карточку. Если это была последняя карточка в текущей очереди, на экране появится сообщение «Нет карточек для повторения».
5. **Сброс прогресса**
   1. Внизу страницы находится кнопка **«Сбросить прогресс»**. При нажатии все интервалы обнуляются, localStorage очищается, и карточки становятся вновь доступными к изучению.

##### 3.3 Особенности и советы

* Если вы изучаете тему впервые, все карточки будут с интервалом 0, и вы сможете пройти их подряд.
* Для максимальной эффективности повторяйте слова сразу же после «Помню», когда приложение предлагает «Следующий повтор через 1 минуту».
* Озвучивание особенно полезно при подготовке к устным экзаменам или собеседованиям.
* При длительном перерыве запустите приложение заново, и те слова, для которых nextReview уже наступил, будут автоматически доступны.

### 4 ТЕХНИЧЕСКОЕ ОПИСАНИЕ ДЕТАЛЕЙ РЕАЛИЗАЦИИ

##### 4.1 Общая архитектура

Приложение построено по классической клиент-серверной модели. Клиент­ская часть отвечает за отображение интерфейса, обработку кликов и визуальную логику, а серверная часть — за работу с данными, хранение JSON и обеспечение целостности прогресса.

* **Клиент** загружает темы и карточки через Fetch API, сохраняет локальные данные в localStorage для ускорения доступа, реагирует на действия пользователя мгновенно, не дожидаясь ответа сервера, если информация уже в кеше.
* **Сервер** хранит данные слов (cards.json), обрабатывает GET и PUT запросы, обновляет прогресс в файле и поддерживает CORS.

##### 4.2 Структура проекта

/backend  
 ├─ main.cpp // точка входа, настройка маршрутов httplib  
 ├─ cards.cpp // класс для загрузки и сохранения JSON  
 └─ include/  
 ├─ cards.hpp   
 ├─ json.hpp   
 └─ httplib.h  
  
/frontend  
 ├─ index.html // главная страница с кнопкой «Начать»  
 ├─ cards.html // страница с интерфейсом изучения слов  
 ├─ css/  
 │ └─ styles.css // стили оформления  
 └─ js/  
 ├─ main.js // логика главной страницы  
 └─ cards.js // логика страницы изучения слов

##### 4.3 Формат и содержание cards.json

Файл содержит объект с ключом "topics". Значением является объект, в котором каждая тема — это массив карточек. Одна карточка — JSON-объект со следующими полями:

* id (integer) — уникальный идентификатор карточки в теме;
* word (string) — английское слово;
* translation (string) — русский перевод;
* progress (object) — содержит:
  + interval (integer) — текущий интервал повторения в минутах;
  + nextReview (integer или null) — UNIX-метка времени, когда карточка станет доступна вновь.

##### 4.4 Тестирование

* **Функциональное тестирование**: вручную проверяли каждый пользовательский сценарий: выбор тем, ответы на карточки, отображение статистики, сброс прогресса.
* **Кросс-браузерное тестирование**: успешно проверено в Chrome, Firefox, Edge.
* **Тестирование озвучивания**: Web Speech API корректно произносит английские слова при скорости речи 0.9 и высоте голоса 1.

### 5 СБОРКА И РАЗВЁРТЫВАНИЕ

##### 5.1 Требования к ПО

* **Серверная часть**:
  + Компилятор C++17 (g++ ≥ 7.0 или MSVC 2019+).
  + Библиотеки httplib и nlohmann::json (заголовочные файлы).
* **Клиентская часть**:
  + Любой современный браузер с поддержкой ES6 и Web Speech API.

##### 5.2 Сборка и запуск

1. Перейдите в каталог и запустите build.bat (build.sh для MacOS).

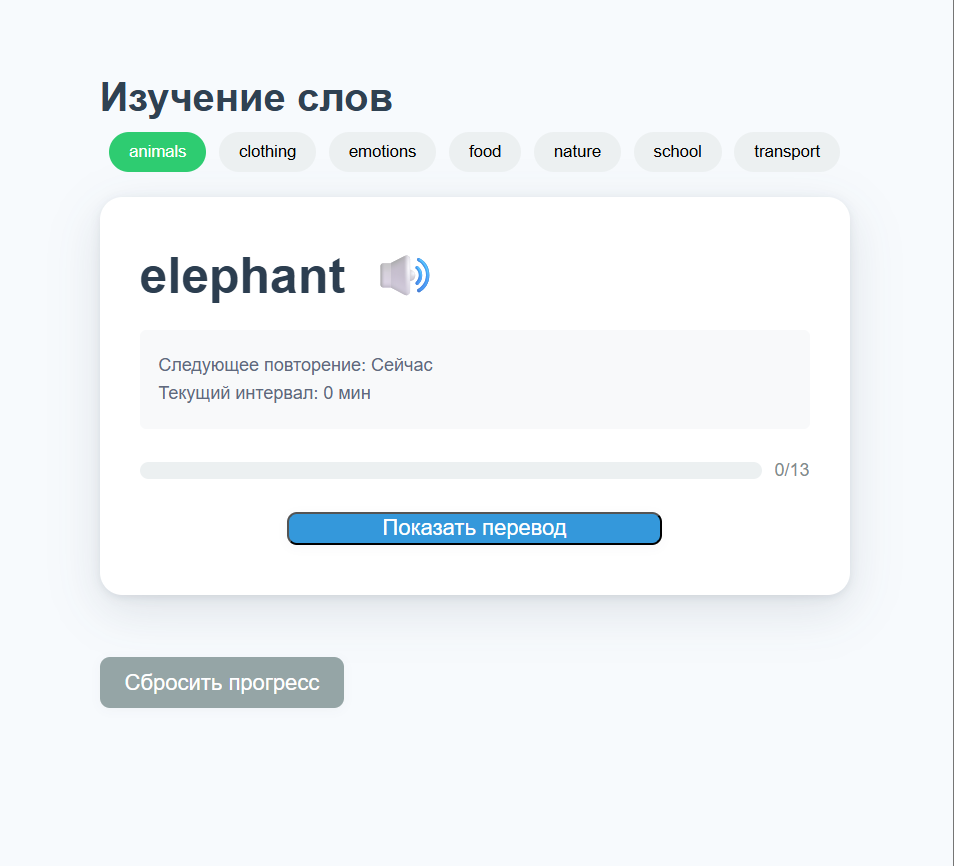
Сервер начнёт слушать запросы по адресу localhost:8080.

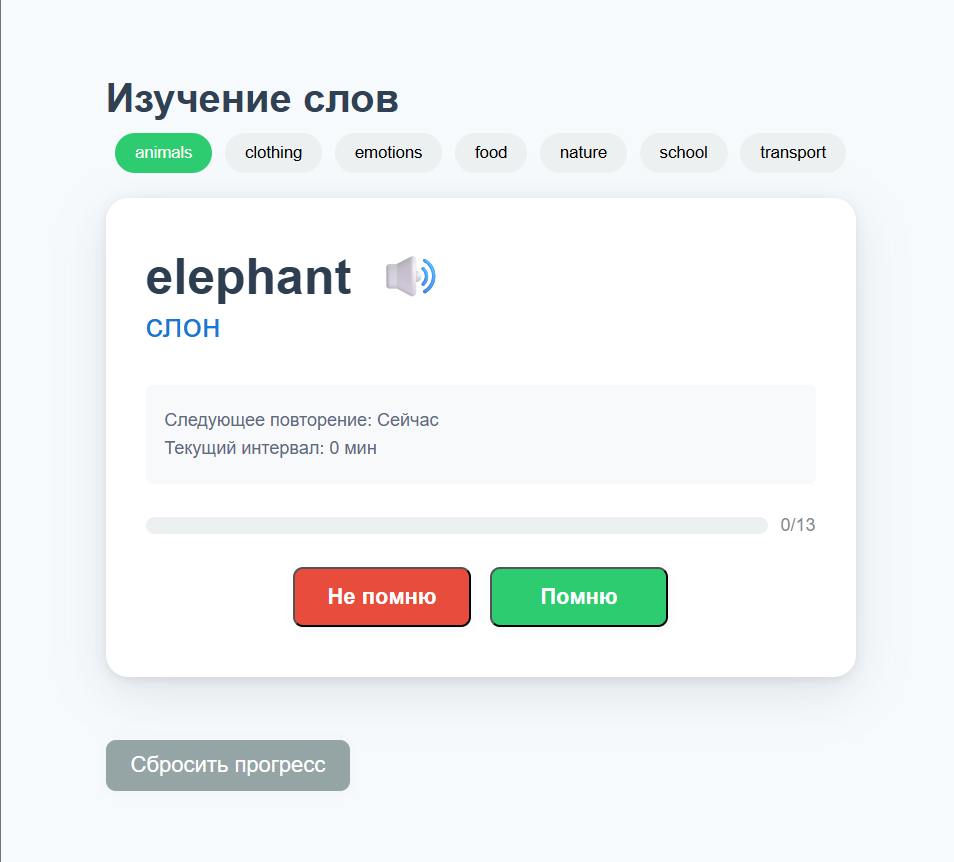
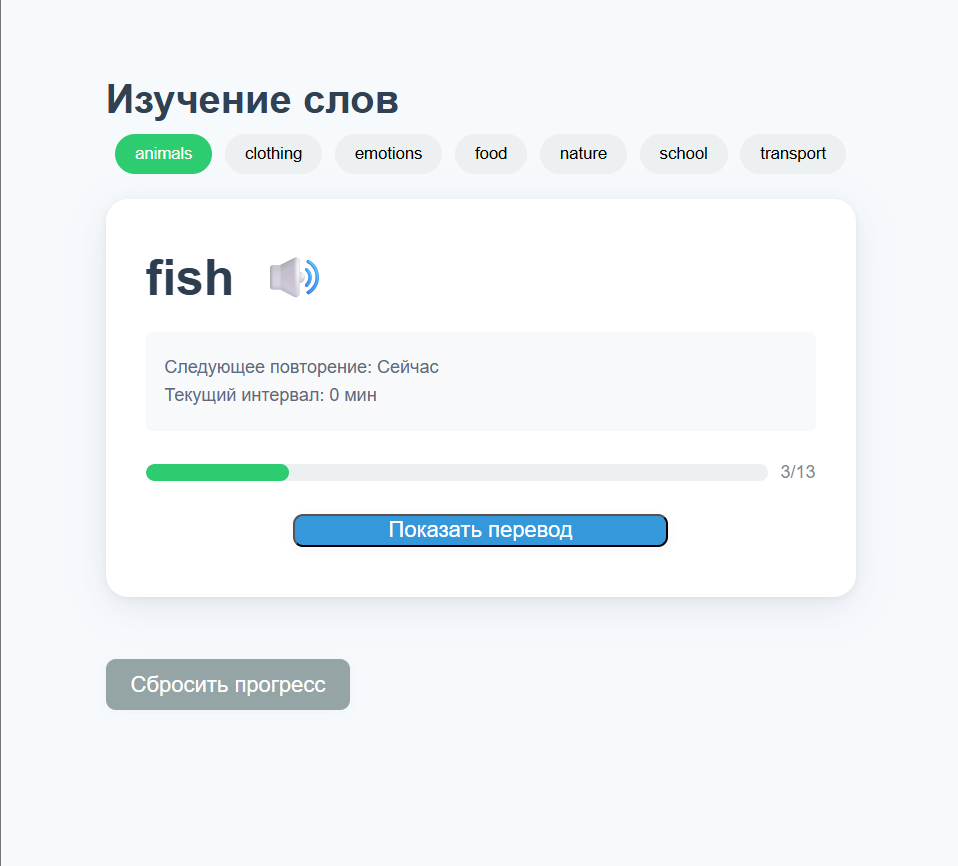
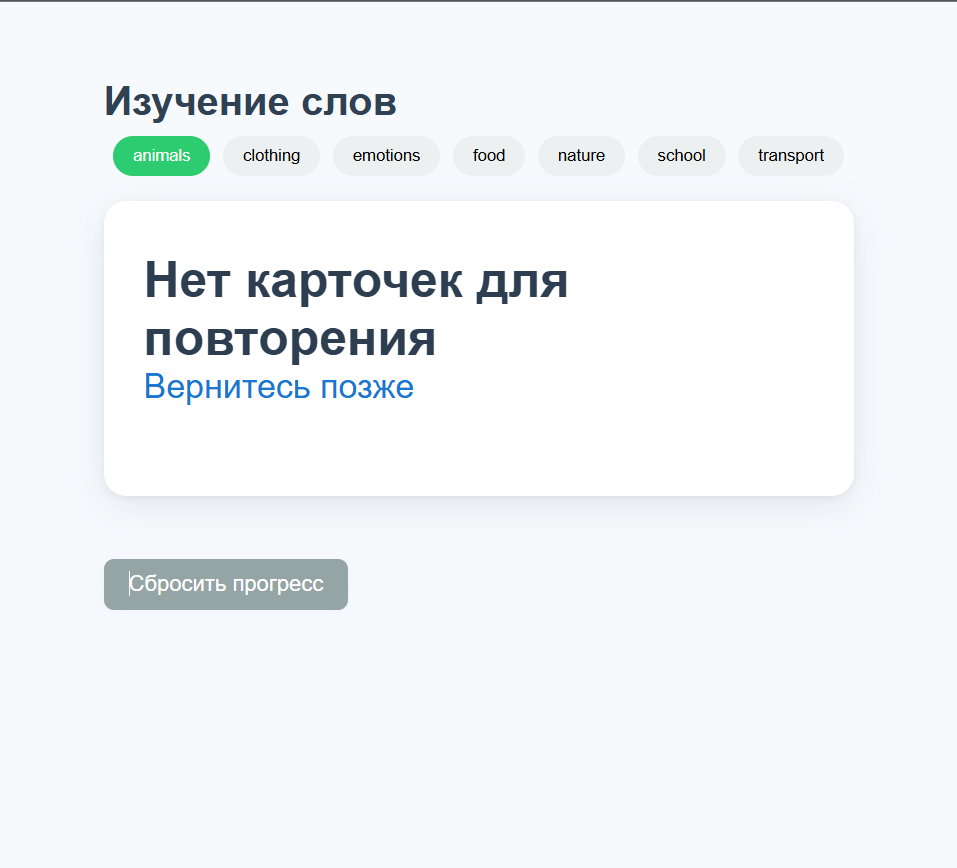
1. Перейдите в папку frontend и откройте index.html в браузере.

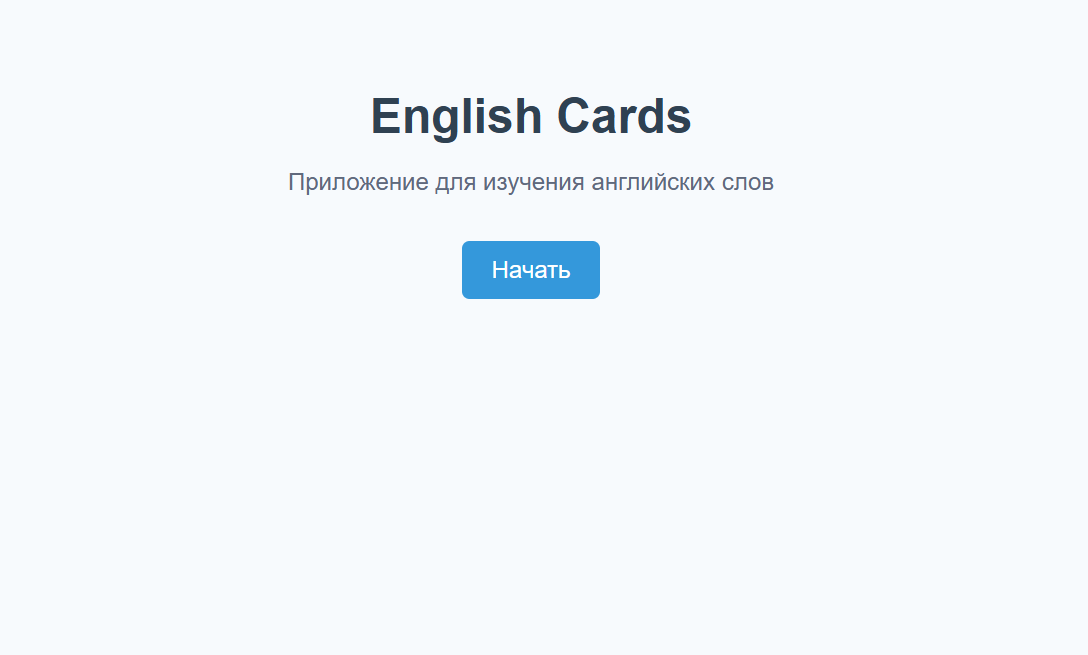
### 6 ПРИЛОЖЕНИЯ

##### Приложение А. Скриншоты интерфейса

* **Рисунок А.1** — Страница изучения: карточка до перевода (слово на английском + значок динамика).



* **Рисунок А.3** — Карточка после нажатия «Показать перевод» (отображён перевод и кнопки «Помню»/«Не помню»).
* **Рисунок А.4** — Прогресс-бар и информация об интервале повторения.
* **Рисунок А.5** — Сообщение «Нет карточек для повторения» при окончании темы.
* **Рисунок А.6** — Главная страница с кнопкой «Начать».



### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация cpp-httplib — GitHub.
2. Документация nlohmann/json — GitHub.
3. MDN Web Docs. [Web Speech API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API).