

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 23.Б09-мм

Создание веб-приложения для изучения ЯПОНСКОГО ЯЗЫКА

Киселев Сергей Александрович

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
ст. преподаватель кафедры системного программирования Зеленчук И. В.

Санкт-Петербург
2025

Оглавление

Введение	4
1. Постановка задачи	5
2. Обзор	6
2.1. Обзор существующего решения	6
2.2. Процесс обучения	6
2.3. Страница изучения новых предметов	6
2.4. Страница повторения	7
2.5. База данных предметов	7
2.6. Обзор технологий для реализации	8
2.7. Выводы	10
3. Описание решения	12
3.1. Архитектура	12
3.2. Архитектура серверной части	12
3.3. Архитектура клиентской части	13
3.4. Типовой сценарий использования приложения	13
3.5. Макет интерфейса	14
4. Описание реализации	16
4.1. Пользовательские аккаунты	16
4.2. Главный экран	17
4.3. Список предметов уровня	18
4.4. Страница предмета	19
4.5. Страница поиска	21
4.6. Страница профиля	22
4.7. Интерфейс обучения	23
4.8. Интерфейс повторения	24
5. Апробация	27
Заключение	28

Введение

Японский язык включает три основных письменности: хирагану, катану и кандзи — заимствованные из китайского языка иероглифы. Большая часть трудностей в изучении языка исходит из системы кандзи, так как она сильно отличается от алфавитных языков. Министерство образования Японии выделило список из около двух тысяч самых часто используемых иероглифов которые нужно знать чтобы свободно владеть языком [13].

Существуют различные способы изучения кандзи — заучивание, на основе мнемоник и контекстуальный. Использование мнемоники при изучении кандзи подразумевает использование ключевых слов для обозначения отдельных компонентов кандзи. Соединяя эти ключевые слова в предложение, учащиеся получают полезный инструмент для запоминания кандзи. Использование мнемоник в качестве стратегии изучения кандзи набирает популярность. Исследования показывают, что использование этого метода помогает как в обучении, так и в извлечении смысла из кратковременной памяти [12].

Существует популярное приложение Wanikani [5], основанное на методике, представленной в пособии "Remembering The Kanji" [1, 12], которое предлагает мнемонические техники для запоминания иероглифов. Для обеспечения эффективного запоминания в долговременной памяти Wanikani использует систему интервального повторения (Spaced Repetition System, SRS) [18], которая помогает пользователям повторять изученные кандзи с оптимальными интервалами для закрепления знаний.

Суть данной работы заключается в создании бесплатного веб-приложения для изучения кандзи на примере реализации Wanikani. В отличие от Wanikani, новое приложение будет бесплатно, поддерживать русский язык и использовать альтернативную реализацию системы интервального повторения.

1. Постановка задачи

Целью работы является создание веб-приложения для изучения японского языка на основе мнемоник. Для её выполнения были поставлены следующие задачи.

1. Рассмотреть существующую реализацию приложения для изучения иероглифов с мнемониками
2. Выполнить обзор технологий для создания веб-приложения.
3. Сформулировать требования к приложению.
4. Разработать макет интерфейса приложения.
5. Создать MVP¹, взяв существующее приложение как пример реализации.
6. Провести апробацию MVP.

¹MVP (англ. minimum viable product, минимально жизнеспособный продукт) — продукт, обладающий минимальными, но достаточными для удовлетворения первых потребителей функциями.

2. Обзор

2.1. Обзор существующего решения

Wanikani.com [5] — это веб-сервис для изучения японского языка, предлагающий основанный на Remembering The Kanji подход к изучению иероглифов. На каждом этапе обучения пользователю предоставляются мнемонические истории, которые помогают лучше запомнить значения и чтения через ассоциации. Обзор был выполнен на основе гайда Wanikani [6, 3, 4].

2.2. Процесс обучения

Обучение на Wanikani состоит из двух основных этапов:

1. **Изучение новых предметов:** Пользователю представляется новый материал, сопровождаемый мнемоникой, помогающей запомнить значение и чтение.
2. **Повторение:** Пользователь повторяет изученный материал, чтобы закрепить знания. Система SRS определяет, когда именно материал должен быть повторен, основываясь на предыдущих ответах.

2.3. Страница изучения новых предметов

- Новые предметы представляются поочередно.
- К каждому предмету прилагается готовая мнемоника.
- Пользователь может добавить свою собственную мнемонику.
- Количество предметов для изучения за один раз устанавливается в профиле пользователя (обычно около 5).

- Страница предоставляет возможность просматривать значение, чтение, мнемоники и примеры использования для каждого нового предмета.

2.4. Страница повторения

- Пользователю предлагается ответить на вопросы по уже изученному материалу.
- Если ответ неверен, материал будет показан снова через более короткий промежуток времени.
- Повторение организовано с использованием SRS — карточки выпадают на повторение в зависимости от прогресса и правильности ответов.
- Страница предлагает пользователю вводить ответы на вопросы и предоставляет обратную связь.

2.5. База данных предметов

- Wanikani содержит базу данных со всей информацией по предметам.
- Пользователь может искать и просматривать информацию по любому предмету.
- База данных включает значения, чтения, мнемоники и примеры использования каждого предмета.

Недостатки Wanikani:

- Система интервального повторения, основанная на фиксированных интервалах, может быть неэффективной [20, 2] или подходить не всем пользователям.
- Отсутствие русского языка.

- Высокая стоимость (10\$ В месяц).

2.6. Обзор технологий для реализации

Для разработки веб приложения нужно было выбрать стек технологий, включая фреймворки для фронтенда, бэкенда, интерфейсов, базу данных и алгоритм для реализации системы интервального повторения (SRS). После анализа различных решений был выбран следующий набор технологий:

Next.js [10] — это фреймворк, основанный на React [14], предназначенный для разработки как серверной, так и клиентской части веб-приложений. Ключевыми особенностями Next.js являются высокая производительность, достигаемая за счёт поддержки серверного рендеринга (SSR) и статической генерации страниц (SSG).

Mantine [7] — представляет собой библиотеку React-компонентов, предлагающую обширный набор готовых элементов интерфейса для быстрого создания адаптивных и стильных пользовательских интерфейсов. Mantine идеально сочетается с **Next.js**, предоставляя широкий выбор компонентов и утилит. Для стилизации компонентов Mantine использует CSS Modules.

MongoDB [8] — документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Хранит данные в формате JSON-подобных документов, что обеспечивает гибкость и простоту работы с данными.

Алгоритм SRS: FSRS [19]

Система интервального повторения (англ. spaced repetition system) — это научно обоснованный метод удержания в памяти, заключающийся в повторении запомненного учебного материала по определённым, постоянно возрастающим интервалам. Этот алгоритм часто используется с флеш-картами [16].

Цель алгоритмов SRS — вычислить оптимальный интервал следующего повторения для определённой карты. В современных алгоритмах,

оптимальность интервала определяется тем что вероятность вспомнить карту после его прохождения будет составлять выше требуемого значения, например, 90%.

FSRS (Free Spaced Repetition Scheduler) — это современный открытый алгоритм интервального повторения изначально разработанный для приложения для обучения флеш-картами Anki для замены устаревшего алгоритма SM-2. Сбор статистики показал, что с FSRS пользователю Anki в среднем нужно на 20-30% меньше повторений для запоминания карты [19].

На вход алгоритма подается карта, состоящая из различных внутренних параметров алгоритма, время последнего повторения, время текущего повторения, а результат его работы — предлагаемая следующая дата повторения. Стоит упомянуть что FSRS является свободным алгоритмом повторения (Free) и дата является рекомендуемой — пользователь всё равно может повторить карту раньше или позже.

Различные алгоритмы используют разные методы для получения следующей даты. FSRS отличается использованием 19 параметров в формулах для вычисления следующей даты. Оптимальные параметры FSRS были получены путём оптимизации на датасете из нескольких миллионов повторений 20 тысяч пользователей Anki. Существуют также алгоритмы вроде SM-2 которые не зависят от каких-либо параметров и оптимизации.

Создателями FSRS было проведено исследование производительности различных SRS алгоритмов на датасете из 727 миллионов повторений десяти тысяч пользователей Anki. Выяснилось, что FSRS является одним из лучших алгоритмов, имея при этом малое число параметров [20]. Также, для FSRS существует готовый API для TypeScript TS-FSRS [15] с лицензией MIT и оптимальные параметры полученные на основе большого датасета реальных пользователей.

2.7. Выводы

После анализа существующей платформы для изучения кандзи при помощи мнемонического метода, технологий для реализации, а также алгоритмов интервального повторения, было решено поставить следующие требования к приложению:

2.7.1. Общие требования

- Использование алгоритма интервального повторения FSRS.
- Приложение должно быть написано на стеке технологий, описанном выше.
- В приложении должна быть поддержка русского языка.
- Интерфейс пользователя, удобный для использования на различных устройствах.
- Регистрация пользователей и сохранение прогресса.

2.7.2. Главная страница

- Отображение количества доступных уроков и повторений.
- Кнопки для перехода к страницам уроков, повторений, базе данных, а также профилю.

2.7.3. Страница изучения новых предметов

- Отображение текущего изучаемого предмета.
- Возможность добавления/редактирования пользовательской мнемоники/синонимов.
- Кнопки для навигации между предметами.

2.7.4. Страница повторения

- Отображение текущего предмета для повторения.
- Поле для ввода ответа пользователем.
- Управление повторением с помощью алгоритма SRS.
- Показ верного ответа при неверном ответе пользователя.
- Отображение следующего предмета после ответа на текущий.

2.7.5. База данных

- Поиск предметов по имени или значению.
- Вывод результатов поиска в виде списка.
- Отображение подробной информации по выбранному предмету (значение, чтение, мнемоника, примеры).

3. Описание решения

3.1. Архитектура

В этом разделе будут описана серверная и клиентская архитектура приложения, типовое взаимодействие с пользователем, а также макет интерфейсов клиентской части.

3.2. Архитектура серверной части

Из-за стремления к быстрому прототипированию, было решено использовать монолитную архитектуру в качестве основы серверной части приложения.

Для управления данными была выбрана MongoDB [8], база данных NoSQL, позволяющая гибко работать со структурами данных. Она идеально работает с JSON, поэтому в неё легко загружать данные. Также MongoDB имеет удобный визуальный интерфейс MongoDB Compass и позволяет быстро вносить изменения во время разработки без миграций SQL.

До запуска веб-приложения в базу данных должны быть загружены данные о предметах: мнемоники, значения, отношения, порядок показа и т.д. Данные не поставляются с проектом и пользователь должен сам найти или разработать их. Поддерживается формат данных, схожий с Wanikani.

Серверная часть приложения выполняется на Node.js [11] и использует возможности Next.js с использованием Api Routes. Api Routes позволяют создавать отдельные API-эндпойнты, которые обрабатывают запросы от клиента.

Приложение использует возможности Next.js по серверной и статической генерации страниц. Некоторые страницы приложения рендерятся на стороне сервера (Server-Side Rendering), обеспечивая быструю начальную загрузку и улучшенные показатели производительности. Статическая регенерация ряда страниц повышает скорость доставки содержимого.

3.3. Архитектура клиентской части

Приложение использует Next.js на стороне клиента, реализуя многостраничное приложение с навигацией, похожей на одностраничное. Next.js обеспечивает навигацию между страницами без полной перезагрузки, что создает ощущение динамичного приложения. Интерфейс построен на React [14]: некоторые компоненты рендерятся на стороне клиента, обеспечивая интерактивность. При возникновении интерактивных действий на клиенте, для выполнения функций на сервере используются вызовы к API Routes. Api Routes позволяют обеспечить гибкое взаимодействие между пользовательским интерфейсом и логикой на сервере.

В качестве библиотеки пользовательского интерфейса используется Mantine [7]. Данная библиотека предоставляет большую библиотеку готовых компонентов, что позволяет быстро создавать интерфейсы. Для стилизации компонентов используются CSS Modules, обеспечивающие локальную область видимости стилей и предотвращающие конфликты имен, что упрощает поддержку и масштабирование проекта.

Для аутентификации в приложении применяется NextAuth.js [9], что упрощает процесс авторизации и аутентификации пользователей, обеспечивая готовые решения для работы с различными провайдерами и хранилищами сессий.

Сборка приложения осуществляется при помощи Turbopack [17], что гарантирует быструю и эффективную компиляцию исходного кода.

3.4. Типовой сценарий использования приложения

- Взаимодействие пользователя начинается с регистрации. После регистрации и входа, пользователь переходит на главный экран.
- На главном экране у пользователя есть возможность начать изучение нового материала или повторять предложенные SRS карты.
- После выбора кнопки изучения материала, пользователь переходит на экран где ему предлагают изучить несколько новых предметов.

- После того как пользователь изучил несколько предметов, ему будет предложено выполнить несколько упражнений на повторение пройденного материала. В случае удачного прохождения упражнений, все они будут добавлены в очередь FSRs.
- На главном экране также будет возможность выполнить повторения, когда SRS посчитает нужным сделать их доступными. В случае, если пользователь что-то забыл, ему будет предложено снова посмотреть правильный ответ.
- На сайте также будет присутствовать база знаний с поиском и навигации, которая будет позволять ему в любое время посмотреть интересующую его предметы.

3.5. Макет интерфейса

Для более полного понимания, каким должен быть интерфейс приложения, с помощью сервиса Figma был разработан макет. Он постранично визуализирует структуру приложения, включая расположение элементов управления, внешний вид различных компонентов и их взаимодействие. Создание макета позволило получить четкое представление о том, как приложение будет выглядеть и функционировать.

Рис. 1: Макет интерфейса (часть 1)

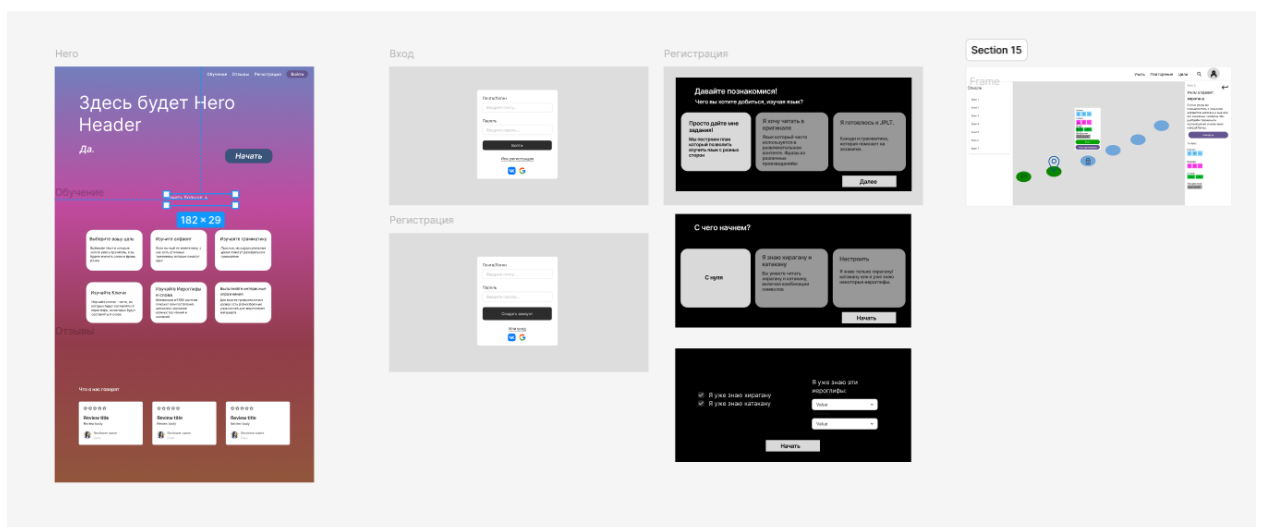
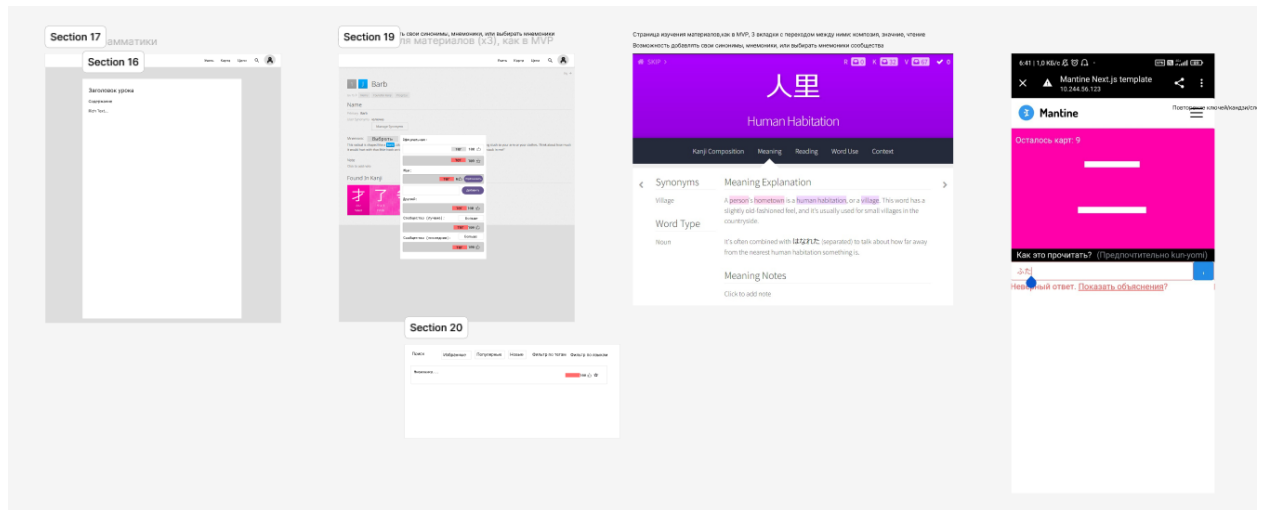


Рис. 2: Макет интерфейса (часть 2)



4. Описание реализации

4.1. Пользовательские аккаунты

Пользовательские аккаунты хранятся в базе данных MongoDB в коллекции users. Для работы с аккаунтами реализованы API маршруты, обеспечивающие регистрацию пользователей.

Для авторизации и аутентификации используется NextAuth который взаимодействует с базой данных.

- Регистрация: Пользователь отправляет данные для регистрации, включая имя, email и пароль. Пароль хэшируется перед сохранением в базе данных. Одновременно с этим для нового пользователя создаются настройки по умолчанию, такие как максимальное количество карт на день и параметры FSRS.
- Вход: Пользователь отправляет свои учетные данные (email и пароль) для входа. NextAuth сравнивает введенный пароль с хэшированным значением, хранящимся в базе данных.

Рис. 3: Форма входа/регистрации

Войти

Создать аккаунт

Имя пользователя

test

Пароль

.....

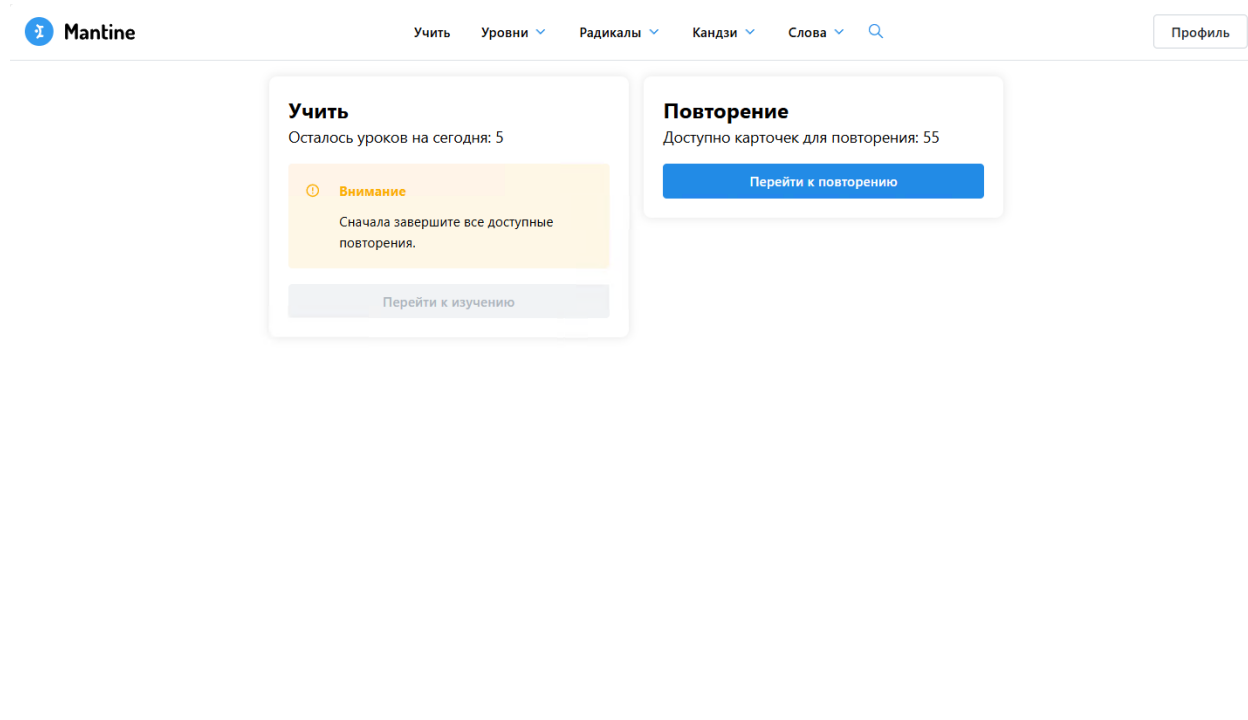
Войти

4.2. Главный экран

После регистрации, пользователь попадает на главный экран.

- В верхнем меню пользователь может видеть различные разделы сайты: обучения, разделы с данными, поиск и профиль.
- В центре пользователь может видеть два виджета - один с новыми уроками и один с повторениями. В день пользователь может получать заданное в профиле число новых уроков. Новые уроки могут быть запущены только после того как пользователь выполнит все доступные повторения.
- Запрос доступного количество урока и повторений реализован с помощью Api Route в серверной части, который извлекает информацию из пользовательского документа.

Рис. 4: Главный экран

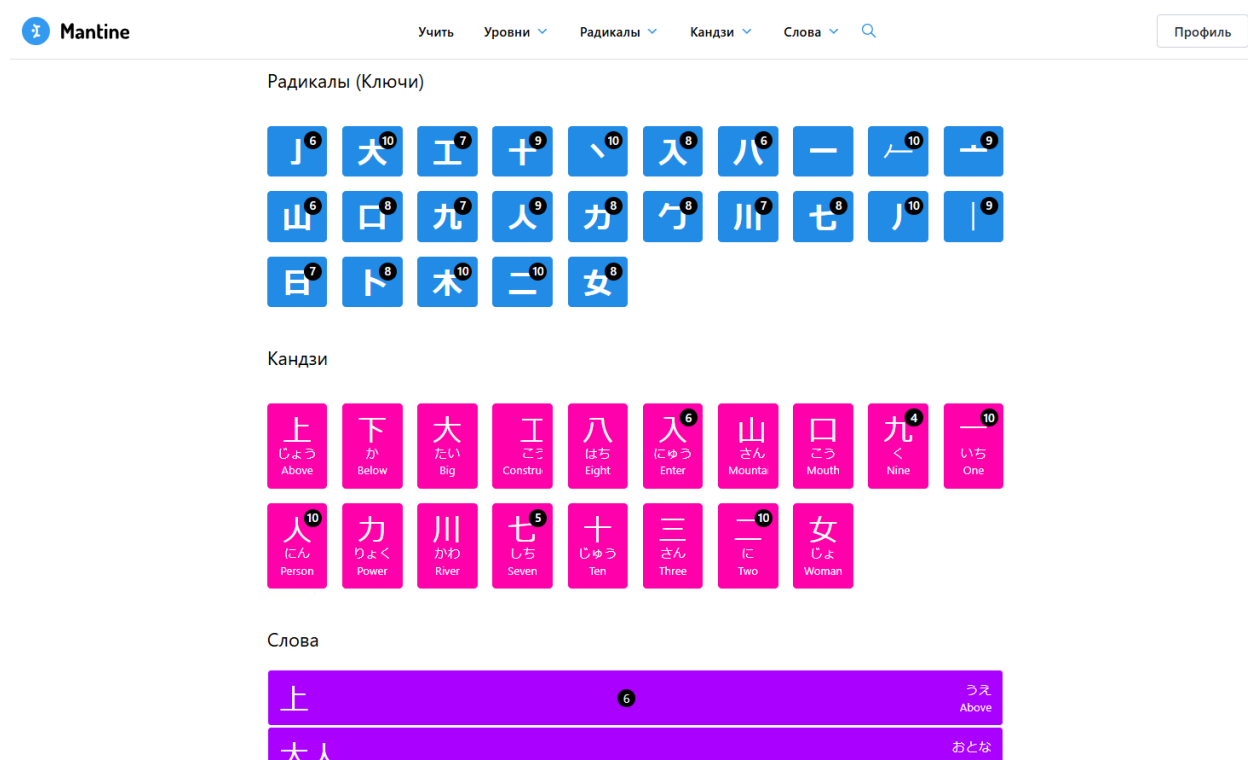


4.3. Список предметов уровня

В списке предметов пользователь может видеть список всех предметов, которые содержит какой-либо уровень. На каждом из предметов есть число, которое показывает сколько раз тот или иной предмет был повторен пользователем. На каждый из предметов можно нажать чтобы открыть страницу с подробностями.

- В серверной части реализованы Api Routes которые позволяют клиентской части запрашивать предметы на конкретном уровне из базы данных.
- Api Route `/stats/` возвращает информацию о том, сколько раз был повторен предмет.
- Кнопки реализованы тремя React компонентами: `Radical`, `Kanji` и `Word`, каждый из которых принимает в аргументы ID предмета и запрашивает информацию о нём (внешний вид, статистику) с серверной части.

Рис. 5: Список предметов уровня



4.4. Страница предмета

У каждого из предметов есть своя страница, на которой есть полезная информация взятая из базы данных, а также написанная пользователем.

- Сверху страницы присутствует описание изучаемого предмета: название, текст, уровень, составные части, значения.
- Ниже, расположен блок с пользовательскими синонимами. Пользователь может самостоятельно добавить свои синонимы или перевод на другой язык. Это реализовано с помощью Api Route на серверной части, который добавляет синонимы пользователя в коллекцию `userSynonyms`. Другой Api Route возвращает уже добавленные синонимы из базы данных.
- Далее расположена информация о мнемониках. Здесь все поля поддерживают выделения текста при помощи XML разметки, что реализовано при помощи отдельного React компонента. Присутствуют поля для добавления собственной мнемоники пользователем или заметок, они реализованы по логике схожей с синонимами, но записывают данные в другую коллекцию (`userMnemonics/userNotes`).
- В конце страницы реализован список предметов, которые используют этот предмет. Информация об этом берётся из базы данных и отображается схожим образом с тем как это делается в списке всех предметов.

Рис. 6: Страница предмета

 Mantine

Учить
 Уровни
 Радикалы
 Кандзи
 Слова

Профиль

上 - Above

Кандзи

Уровень: 1

Комбинация радикалов


 +
 

Значения

Above (основное)
 Up
 Over

Пользовательские синонимы

d
 s

Добавить синоним

Добавить

Мнемоника значений

Рис. 7: Мнемоники предмета

 Mantine

Учить
 Уровни
 Радикалы
 Кандзи
 Слова

Профиль

Мнемоника значений

You find a **toe** on the **ground**. It's weird, because it's **above** the ground, not where toes belong.

Подсказка
 Focus your imagination on how body parts should be buried, not above ground like this. Sure, finding a toe is a little strange, but finding a toe above the ground? Downright silly.

Пользовательская мнемоника значений

Чтения

On'yomi: **じょう**

Kun'yomi: **うえ** **あ** **のぼ** **うわ** **かみ**

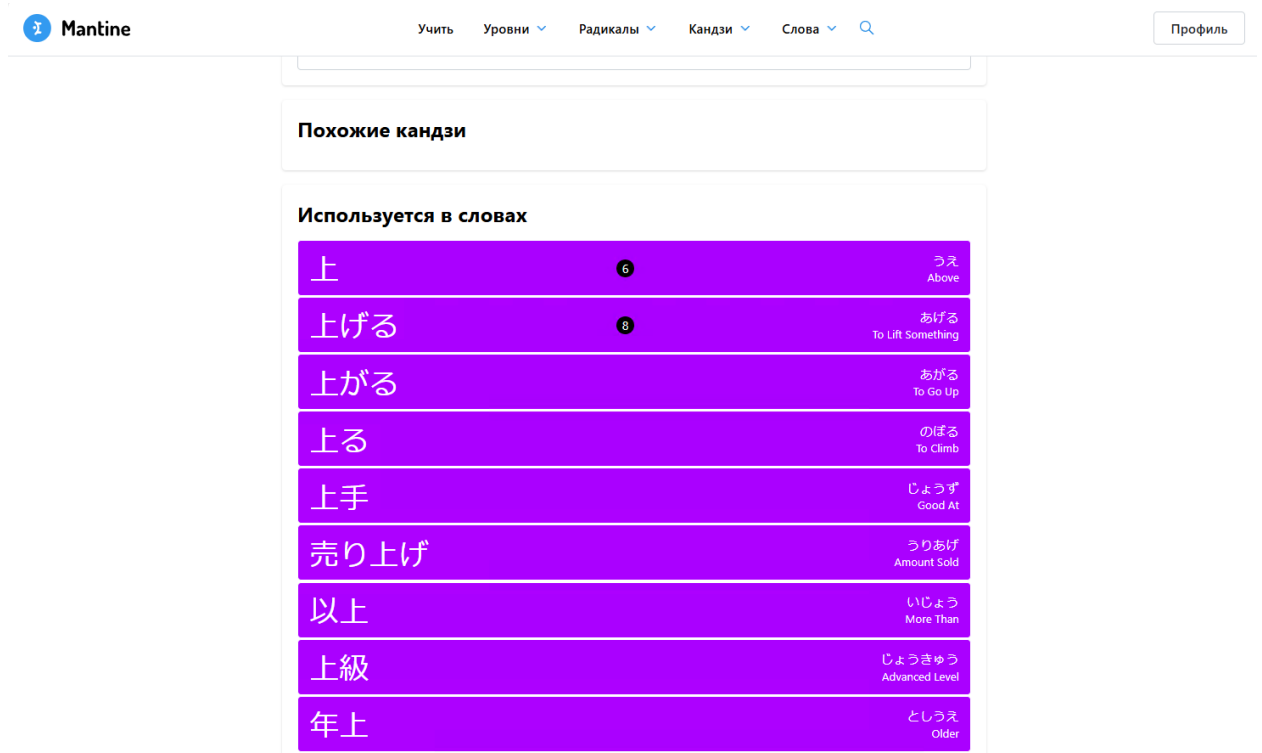
Nanori:

Мнемоника чтений

Of course when you find a toe **above** the ground, you want to know where the toe came from. When you examine it, you see a name written on there. This toe belongs to the local clumsy farmhand, **Joe** (**じょう**).

Подсказка
 Try to remember Joe, the local farmhand. He is a recurring character that will come up anytime the reading for a kanji is じょう. He is a very big man, and he wears farmer's

Рис. 8: Отношения предмета

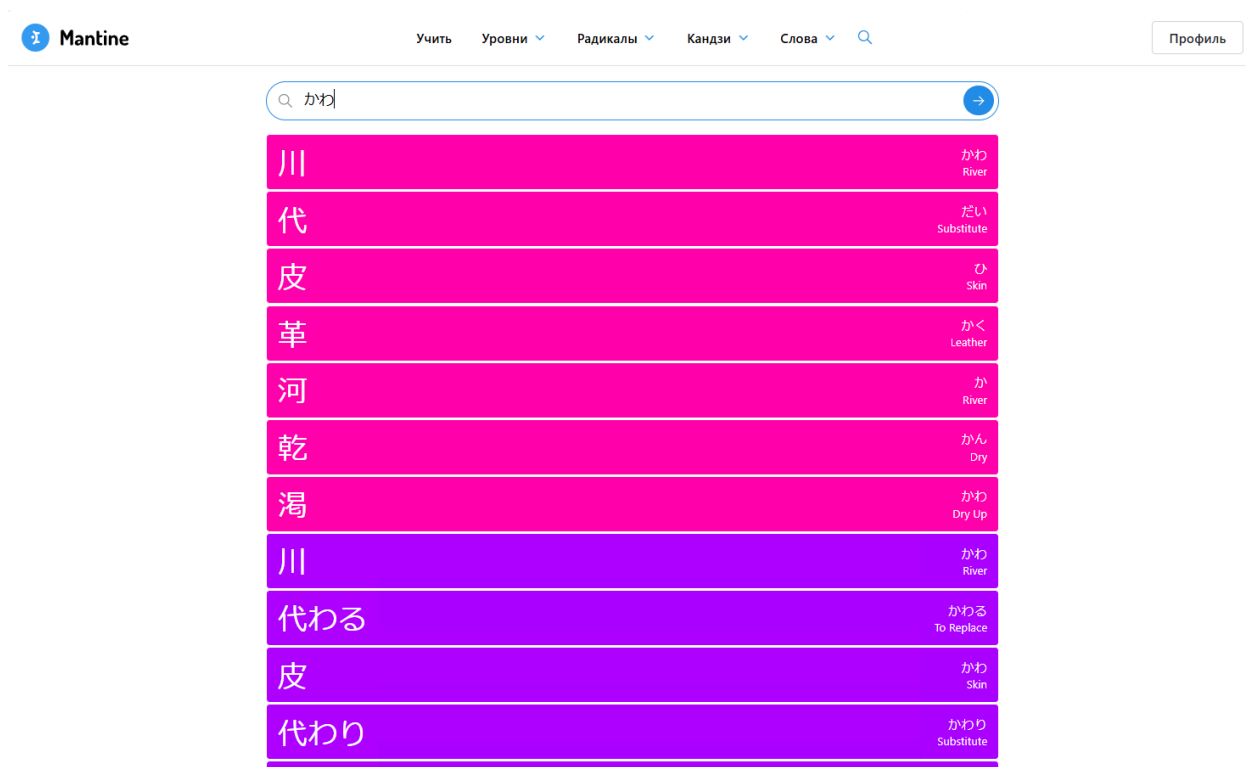


4.5. Страница поиска

На странице поиска пользователь может найти любой предмет по его имени, значению или произношению.

- На странице присутствует строка поиска, в которую пользователь может ввести любой запрос. Нажатие клавиши Enter или иконки поиска выдаёт новые результаты.
- Со стороны сервера поиск реализован с помощью Api Route, который принимает запрос пользователя, ищет по всем документам в базе данных и возвращает список предметов.
- Отображение найденных предметов реализовано теми же React компонентами, что и в списке предметов уровня.

Рис. 9: Поиск



4.6. Страница профиля

На странице профиля пользователь может изменить свои настройки или выйти из аккаунта

- Поле максимальных предметов в день регулирует количество предметов которые пользователь может получить в день.
- Поле количество предметов за раз регулирует число предметов которые пользователь будет изучать за одно нажатие кнопки «Учить» в главном меню.
- Задание и получение настроек реализовано при помощи Api Routes на стороне сервера которые записывают и получают данные из документа пользователя.

Рис. 10: Профиль

Максимум уроков в день

15

Уроков в одной пачке

5

Сохранить настройки

Выйти из профиля

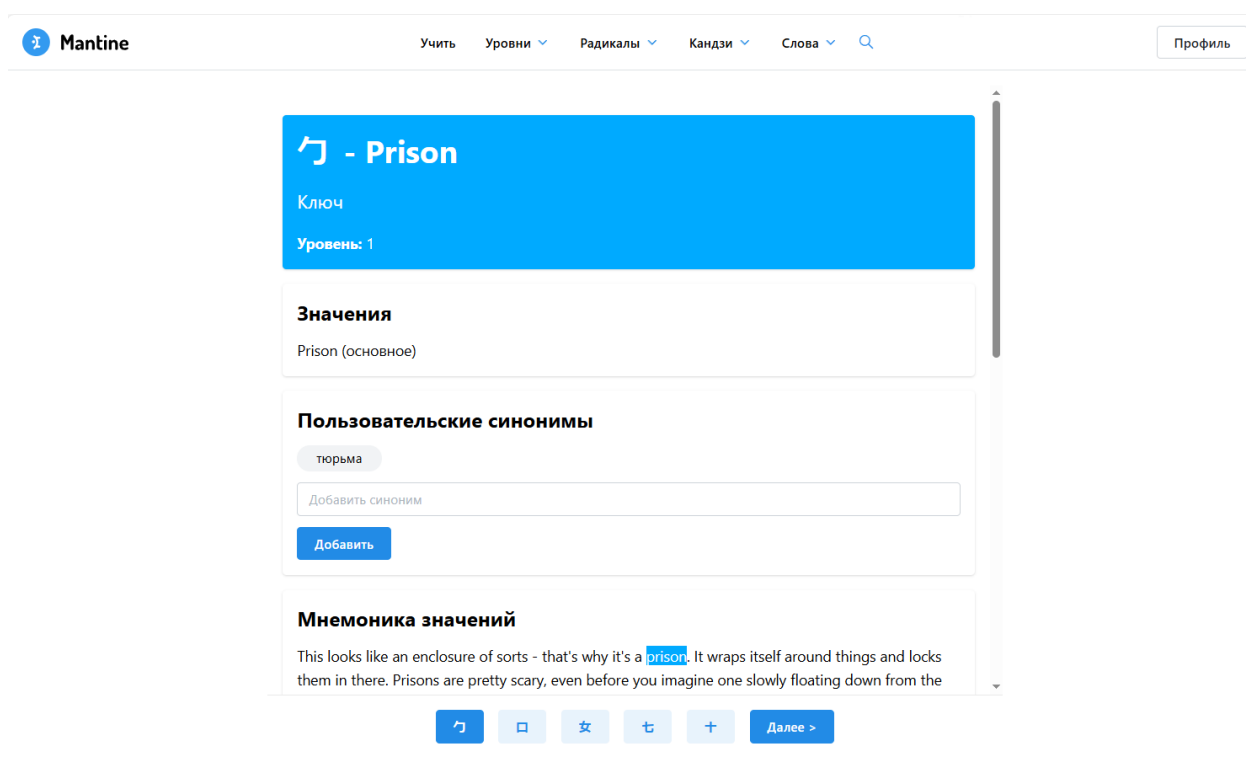
4.7. Интерфейс обучения

Когда пользователь решает что он готов приступить к обучению, он должен нажать кнопку «Учить» в главном меню. На экране обучения, ему предлагается посмотреть по очереди определенное число предметов и внимательно ознакомиться со всем связанным с ними материалом. Пользователь может добавить свои синонимы или переводы при помощи соответствующих действий. Мнемоники помогут пользователю лучше запомнить материал. Если ему не понравились предложенные мнемоники, он может добавить свои.

- На серверной части присутствуют Api Routes которые контролируют какие предметы будет получать пользователь в новых уроках. Из всех коллекций базы данных достаются предметы, сортируются по порядку в уроке и нужное число из них возвращается пользователю. Если число уроков превышает максимум за день, то новые уроки не будут выданы.

- После того как пользователь прошел уроки, его автоматически перебрасывает на упражнения с повторением, для того чтобы закрепить материал. В этот же момент вызывается другой Api Route, который записывает в базу данных сдвиг позиции пользователя на конкретном уровне. Если пользователь прошел все предметы на одном уровне, он переходит на следующий. Также, для каждого из изученного предметов создаются карточки FSRS и добавляются в очередь.

Рис. 11: Профиль



4.8. Интерфейс повторения

В интерфейс повторения пользователь может попасть либо если он только что изучил новый материал, либо если пришел срок повторения ранее изученных карт FSRS. Здесь пользователь может последовательно попытаться вспомнить каждую из предложенных FSRS карт. В случае неудачи, пользователю будет предложено ещё раз посмотреть на предмет, а алгоритм FSRS учтёт это при планировании следующих повторений.

- На серверной части Api Route проверяет все предметы у которых истек срок повторения и возвращает их.
- Ещё один Api Route получает результат повторения конкретной карты и применяет к ней алгоритм FSRS.
- Также реализована проверка, если пользователь перепутал чтение иероглифа с чтением слова которое записывается также, ему будет дан второй шанс исправить ответ.

Рис. 12: Повторение

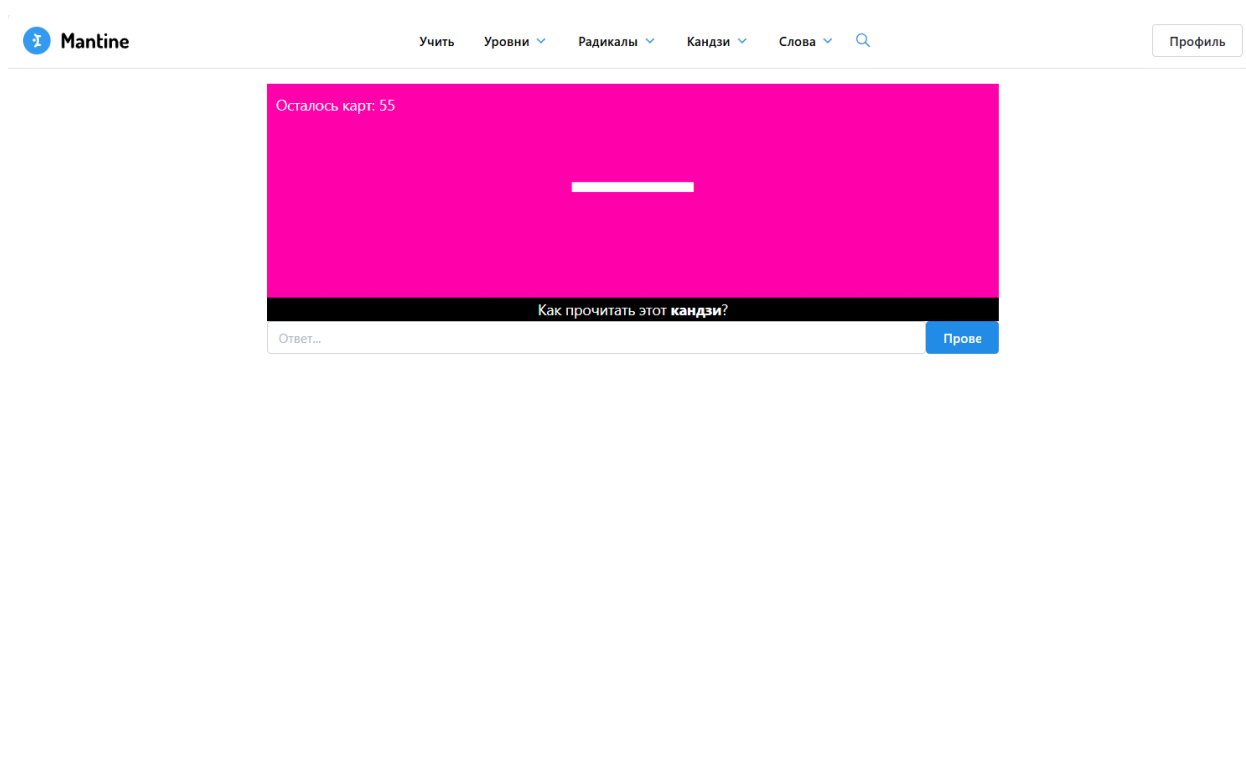


Рис. 13: Повторный показ материала при ошибке

 Mantine

Учить Уровни ▾ Радикалы ▾ Кандзи ▾ Слова ▾ 🔍

Профиль

Осталось карт: 54

上

Что означает этот **слово**?

под

Далее

Неверный ответ. [Показать объяснения?](#)

上 - Above

Слово

Уровень: 1

Часть речи: noun suffix

Значения

Above (основное) Up Over

5. Апробация

Апробация проводилась с использованием шкалы System Usability Scale (SUS), предназначенной для оценки удобства использования систем, интерфейсов или продуктов. Шкала включает 10 вопросов, на которые пользователи дают ответы, оценивая их по шкале от 1 до 5, где 1 означает полное несогласие, а 5 — полное согласие. Полученные оценки суммируются и преобразуются в итоговый SUS-балл в диапазоне от 0 до 100. В ходе апробации было опрошено 10 пользователей, а вычисленный итоговый балл составил 72.5.

В ходе апробации были получены следующие комментарии по работе MVP:

- Просьба добавить тёмную тему, чтобы глазам было проще
- Просьба добавить больше статистики о каждом предмете, а также о проценте прохождения курса
- Просьба добавить рабочие ссылки на страницы предметов из мнемоник

Заключение

В итоге были достигнуты следующие результаты.

- Было рассмотрено существующее приложение для обучения японскому языку мнемониками.
- Был проведен обзор технологий для создания веб-приложения
- Были сформулированы требования к приложению.
- Был разработан макет интерфейса приложения.
- Был создан MVP, соответствующий требованиям.

Код доступен по ссылке <https://github.com/anunknowperson/jrs>

- Проведена апробация MVP по шкале SUS, итоговый балл составил 72.5.

Список литературы

- [1] Heisig James W. Remembering the Kanji 1: A Complete Course on How Not to Forget the Meaning and Writing of Japanese Characters // Remembering the Kanji 1: A Complete Course on How Not to Forget the Meaning and Writing of Japanese Characters. — University of Hawaii Press, 2011.
- [2] LLC Tofugu. WaniKani's SRS Stages. — URL: <https://knowledge.wanikani.com/wanikani/srs-stages/> (дата обращения: 21 декабря 2024 г.).
- [3] LLC Tofugu. Wanikani - First Lessons. — URL: <https://knowledge.wanikani.com/getting-started/first-lessons/> (дата обращения: 21 декабря 2024 г.).
- [4] LLC Tofugu. Wanikani - First Reviews. — URL: <https://knowledge.wanikani.com/getting-started/first-reviews/> (дата обращения: 21 декабря 2024 г.).
- [5] LLC Tofugu. Wanikani - SRS для изучения иероглифов и слов Японского Языка. — URL: <https://wanikani.com> (дата обращения: 21 декабря 2024 г.).
- [6] LLC Tofugu. What is Wanikani. — URL: <https://knowledge.wanikani.com/getting-started/how-wanikani-works/> (дата обращения: 21 декабря 2024 г.).
- [7] Mantine. — URL: <https://mantine.dev/> (дата обращения: 21 декабря 2024 г.).
- [8] MongoDB. — URL: <https://www.mongodb.com/> (дата обращения: 21 декабря 2024 г.).
- [9] NextAuth.js. — URL: <https://next-auth.js.org/> (дата обращения: 21 декабря 2024 г.).

- [10] Next.js. — URL: <https://nextjs.org/> (дата обращения: 21 декабря 2024 г.).
- [11] Node.js. — URL: <https://nodejs.org/> (дата обращения: 21 декабря 2024 г.).
- [12] Paxton Simon, Svetenant Chavalin. Tackling the Kanji hurdle: Investigation of Kanji learning in Non-Kanji background learners // [International Journal of Research Studies in Language Learning](#). — Vol. 3, no. 2.
- [13] Rasiban Linna Meilia, Sudana Dadang, Sutedi Dedi. Indonesian students' perceptions of mnemonics strategies to recognize Japanese kanji characters // [Indonesian Journal of Applied Linguistics](#). — Vol. 8, no. 2. — URL: <https://ejournal.upi.edu/index.php/IJAL/article/view/15256>.
- [14] React. — URL: <https://react.dev/> (дата обращения: 21 декабря 2024 г.).
- [15] Repetition Open Spaced. TS-FSRS. — URL: <https://github.com/open-spaced-repetition/ts-fsrs> (дата обращения: 21 декабря 2024 г.).
- [16] Smolen Paul, Zhang Yili, Byrne John H. The right time to learn: mechanisms and optimization of spaced learning // [Nature Reviews Neuroscience](#). — Vol. 17, no. 2.
- [17] Turbopack. — URL: <https://turbo.build/pack> (дата обращения: 21 декабря 2024 г.).
- [18] Wozniak Piotr. Optimization of learning. — URL: <https://super-memory.com/english/ol.htm> (дата обращения: 21 декабря 2024 г.).
- [19] Ye Jarrett et al. FSRS - Modern Spaced Repetition Algorithm. — URL: <https://github.com/open-spaced-repetition/fsrs4anki> (дата обращения: 21 декабря 2024 г.).

- [20] Сравнение производительности различных алгоритмов SRS. — URL: <https://github.com/open-spaced-repetition/srs-benchmark> (дата обращения: 21 декабря 2024 г.).