Japanese Learning Web App

Technical Plan

# Overview

This document outlines the technical specifications for the Japanese Learning Web App. The app aims to help users learn Japanese through flashcards, practice sessions and an AI-powered sentence corrector. It also tracks user progress and uses spaced repetition for memory retention.

### **Tech Stack:**

* **Frontend**: Next.js (for dynamic user interface and SSR)
* **Backend**: Node.js, AWS Lambda, API Gateway
* **Database**: AWS RDS (PostgreSQL)
* **AI Integration**: OpenAI API for sentence correction
* **Hosting**: AWS Lambda (for serverless functions)
* **Authentication**: JWT-based authentication

# API Endpoints

1. User Authentication

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endpoint** | **Method** | **Description** | **Input** | **Output** |
| /register | POST | Register a new user | {email, username, password} | {user\_id, token} |
| /login | POST | Log in an existing user | {email/username, password} | {user\_id, token} |
| /logout | POST | Log out a user | {token} | {message} |

1. Flashcards

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endpoint** | **Method** | **Description** | **Input** | **Output** |
| /flashcard/lesson | GET | Fetches flashcard content | {user\_id} | {flashcards: […]} |
| /flashcard/lesson | POST | Marks flashcard as known or unknown | {user\_id, flashcard\_id, familiarity} | {status} |
| /flashcards/review | GET | Fetches flashcards ready for review | {user\_id} | {flashcards: […]} |
| /flashcards/review | POST | Marks a flashcard as reviewed | {user\_id, flashcard\_id, correct} | {status} |

1. Practice Session

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endpoint** | **Method** | **Description** | **Input** | **Output** |
| /practice | GET | Fetch practice question | {user\_id, practice\_type} | {practice\_exercise} |
| /practice | POST | Submit answer | {user\_id, answer} | {status} |
| /practice/breakdown | POST | Breaks down a mistake | {user\_sentence, correct\_sentence} | {correct\_senetence, explanation} |

1. AI Sentence Correction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endpoint** | **Method** | **Description** | **Input** | **Output** |
| /correct | POST | Corrects a Japanese sentence | {user\_sentence} | {corrected\_sentence, explanation} |
| /breakdown | POST | Breaks down a mistake | {user\_sentence, correct\_sentence} | {correct\_sentence. explanation} |

1. Progress Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endpoint** | **Method** | **Description** | **Input** | **Output** |
| /progress | GET | Retrieves user progress | {user\_id} | {progress\_data} |
| /progress | POST | Updates progress data | {user\_id, progress\_data} | {status} |

1. Pricing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Endpoint** | **Method** | **Description** | **Input** | **Output** |
| /pricing | POST | Submit selected subscription | {user\_id, subscription} | {status} |

# Database Schema

1. Users

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| user\_id | UUID | Primary key |
| username | VARCHAR (50) | Username |
| email | VARCHAR (255) | User email |
| password\_hash | CHAR | Encrypted password |
| subscription\_tier | VARCHAR (50) | Free, basic, premium |

1. Flashcards

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| flashcard\_id | UUID | Primary key |
| type | VARCHAR (10) | Type of flashcard (kanji/vocab/grammar) |
| content | JSONB | Flashcard data (see below) |
| breakdown | JSONB | Example sentence breakdown |
| JLPT | VARCHAR (10) | Flashcard difficulty |
| audio\_url | JSONB | url to audio files in s3 bucket |

1. UserFlashcards

Flashcard data JSON:

{

“flashcard”: …,

“meaning”: …,

“reading”: [

{“onyomi”: …, “kunyomi”: …}

],

“examples”: [

{“sentence”: …, “translation”: …}

]

}

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | UUID | Primary key |
| user\_id | UUID | Foreign key (Users) |
| flashcard\_id | UUID | Foreign key (Flashcards) |
| level | INTEGER | Indicator of how well a user knows a flashcard |
| correct\_count | INTEGER | Correctly answered count |
| incorrect\_count | INTEGER | Incorrectly answered count |
| next\_review | TIMESTAMP | Next review date |
| accuracy | INTEGER | (correct) / (correct + incorrect) |
| reached\_level\_3 | DATE | Record date when user becomes familiar with a flashcard |

1. Practice

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| practice\_id | UUID | Primary key |
| flashcard\_id | UUID | Foreign key (Flashcards) |
| type | VARCHAR (50) | Type of practice (sentence rearrangement/fill in the blank/audio transcribing) |
| question | VARCHAR (255) | Question |
| answer | VARCHAR (255) | Answer |

1. UserPractice

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | UUID | Primary key |
| practice\_id | UUID | Foreign key (Practice) |
| level | INTEGER | Indicator of how well a user knows the practice |
| correct\_count | INTEGER | Correctly answered count |
| incorrect\_count | INTEGER | Incorrectly answered count |
| accuracy | INTEGER | (correct) / (correct + incorrect) |
| next\_review | TIMESTAMP | Next review date |

1. Progress

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| progress\_id | UUID | Primary key |
| user\_id | UUID | Foreign key (Users) |
| user\_level | INTEGER | User’s JLPT level |
| total\_flashcards | INTEGER | Total flashcards learnt |
| total\_kanji | INTEGER | Total kanji learnt |
| total\_vocab | INTEGER | Total vocab learnt |
| total\_grammar | INTEGER | Total grammar learnt |
| current\_streak | INTEGER | User’s current streak |
| max\_streak | INTEGER | User’s max streak |

# Backend Logic

Key Lambda Functions:

**1. User Authentication Functions**

***LoginUser***

* **Trigger**: API Gateway (POST /login)
* **Description**: Logs in an existing user by verifying credentials and returning a JWT.
* **Input**:
* {
* "email": [user@example.com](mailto:user@example.com) / "username": username,
* "password": "securepassword"
* }
* **Logic**:
  1. **Input Validation**:
     + Check if the email/username and password fields are non-empty.
  2. **Query Database**:
     + Search the Users table for a user matching the provided email.
     + If no user is found, return a 401 Unauthorised error.
  3. **Password Verification**:
     + Use bcrypt to compare the provided password with the hashed password stored in the database.
     + If the password does not match, return a 401 Unauthorised error.
  4. **JWT Token Generation**:
     + Generate a JWT token containing the user's ID and email.
     + Set a token expiration time (e.g., 24 hours).
  5. **Response**:
     + Return the JWT token and the user's ID.
* **Output**:
* {
* "user\_id": "uuid",
* "token": "jwt-token"
* }

**2. Flashcard Management Functions**

***GetLessonFlashcards***

* **Trigger**: API Gateway (GET /flashcards/lesson)
* **Description**: Retrieves flashcards for the user’s lesson.
* **Input**:
* {
* "user\_id`": "uuid"
* }
* **Logic**:
  1. **Input Validation**:
     + Verify that the user\_id is valid and non-empty.
  2. **Database Query**:
     + Query the Flashcards table to fetch:
       - 10 random N5 vocabulary cards.
       - 5 random N5 kanji cards.
       - 3 random N5 grammar points.
     + Ensure these flashcards have not been marked as "learned" by the user.
  3. **Response**:
     + Format the flashcards into a JSON array with relevant fields:
       - word, meaning, reading (if applicable), examples, audio.
* **Output**:
* {
* "flashcards": [
  + {
  + "flashcard": "猫",
  + "meaning": "cat",
  + "reading": "ねこ",
  + "examples": ["猫がいます。"],
  + "audio": "audio\_file\_url"
  + },
  + {
  + "flashcard": "犬",
  + "meaning": "dog",
  + "reading": "いぬ",
  + "examples": ["犬が走ります。"],
  + "audio": "audio\_file\_url"
  + }
* ]
* }

***PostLessonFlashcards***

* **Trigger**: API Gateway (POST /flashcards/lesson)
* **Description**: Submits the user’s familiarity with each flashcard.
* **Input**:
* {
* "user\_id": "uuid",
* "flashcard\_id": "uuid",
* "familiarity": "integer"
* }
* **Logic**:
  1. **Input Validation**:
     + Verify that the user\_id, flashcard\_id, and familiarity values are valid and non-empty.
  2. **Database Query**:
     + Update the UserFlashcard table with:
       - The familiarity level (familiarity value provided by the user).
       - Set next\_review to a short interval if marked "not known".
  3. **Response**:
     + Return a success or failure status.
* **Output**:
* {
* "status": "success"
* }

***GetReviewFlashcards***

* **Trigger**: API Gateway (GET /flashcards/review)
* **Description**: Retrieves flashcards ready for SRS-based review.
* **Input**:
* {
* "user\_id": "uuid"
* }
* **Logic**:
  1. **Input Validation**:
     + Verify that the user\_id is valid and non-empty.
  2. **Database Query**:
     + Query the UserFlashcard table for:
       - Flashcards where next\_review is due.
       - Filter by lesson type (kanji, vocab, or grammar).
  3. **Response**:
     + Format the flashcards into a JSON array with relevant fields:
       - word, meaning, reading (if applicable), examples, audio.
* **Output**:
* {
* "flashcards": [
* {
* "flashcard": "猫",
* "meaning": "cat",
* "reading": "ねこ",
* "examples": ["猫がいます。"],
* "audio": "audio\_file\_url"
* },
* {
* "flashcard": "犬",
* "meaning": "dog",
* "reading": "いぬ",
* "examples": ["犬が走ります。"],
* "audio": "audio\_file\_url"
* }
* ]
* }

***PostReviewFlashcards***

* **Trigger**: API Gateway (POST /flashcards/review)
* **Description**: Submits the user’s answer to flashcard reviews.
* **Input**:
* {
* "user\_id": "uuid",
* "flashcard\_id": "uuid",
* "correct": true
* }
* **Logic**:
  1. **Input Validation**:
     + Verify that the user\_id, flashcard\_id, and correct values are valid and non-empty.
  2. **Database Query**:
     + Update the UserFlashcard table:
       - If the answer is correct:
         * Increase the SRS interval (e.g., from 1 day to 3 days).
       - If the answer is incorrect:
         * Decrease the SRS interval (e.g., reset to 1 day).
  3. **Response**:
     + Return a success or failure status.
* **Output**:
* {
* "status": "success"
* }

**3. AI Sentence Correction Functions**

***CorrectSentence***

* **Trigger**: API Gateway (POST /correct)
* **Description**: Corrects a Japanese sentence using OpenAI API.
* **Input**:
* {
* "user\_sentence": "私はねこがいる。"
* }
* **Logic**:
  1. **Input Validation**:
     + Ensure the input contains a non-empty user\_sentence field.
  2. **OpenAI API Call**:
     + Send the user\_sentence to OpenAI's API for correction.
     + Receive a response containing the corrected sentence and any suggestions.
  3. **Parse Response**:
     + Extract the corrected sentence and explanations from the API response.
  4. **Log Correction**:
     + Insert the original and corrected sentences into the Corrections table along with the explanation and user ID. //maybe
  5. **Response**:
     + Return the corrected sentence and the explanation to the user.
* **Output**:
* {
* "corrected\_sentence": "私はねこがいます。",
* "explanation": "Use the polite form 'います' instead of 'いる'."
* }

**4. Practice Session Functions**

***GetPracticeQuestion***

* **Trigger**: API Gateway (GET /practice)
* **Description**: Retrieves a practice exercise for the user.
* **Input**:
* {
* "user\_id": "uuid",
* "practice\_type": "fill in the gap"
* }
* **Logic**:
  1. **Input Validation**:
     + Ensure user\_id and practice\_type are provided and valid.
  2. **Fetch Question**:
     + Query the PracticeQuestions table to fetch a question matching the practice\_type.
     + Make sure only the currentDate <= next\_review questions are fetched.
  3. **Response**:
     + Format the question and provide the expected answer for validation.
* **Output**:
* {
* "practice\_exercise": {
* "question": "私は猫＿好きです。",
* "correct\_answer": "が"
* }
* }

***PostPracticeQuestion***

* **Trigger**: API Gateway (POST /practice)
* **Description**: Submits user’s answer to the practice question.
* **Input**:
* {
* "user\_id": "uuid",
* "practice\_id": "uuid"
* }
* **Logic**:
  1. **Input Validation**:
     + Ensure user\_id and practice\_id are provided and valid.
  2. **Fetch Question**:
     + Query the PracticeQuestions table to fetch a question matching the practice\_type.
  3. **Response**:
     + Format the question and provide the expected answer for validation.
* **Output**:
* {
* "practice\_exercise": {
* "question": "私は猫＿好きです。",
* "correct\_answer": "が"
* }
* }

# Lesson Structure

N5

* **Lesson 1 and 2**: Introductions & Essentials (2)
* **Lesson 3**: Pronouns, Particles & Movement
* **Lesson 4, 5, 9, 10 and 12**: Descriptive Language & Modifiers (5)
* **Lesson 6**: Family & Relationships
* **Lesson 7 and 8**: Daily Routines & Time (2)
* **Lesson 11**: Actions & Hobbies
* **Lesson 13**: Locations & Directions
* **Lesson 14 and 15**: Counting & Comparisons (2)
* **Lesson 16**: Food & Dining
* **Lesson 17**: Commands & Advice
* **Lesson 18**: Past Experiences
* **Lesson 19**: Permissions & Prohibitions
* **Lesson 20**: Conditionals & Methods
* **Lesson 21**: Polite Speech & Honorifics
* **Lesson 22**: Review & Advanced Topics

Prompt for AI Sentence Breakdown:

**# Prompt 1**

You are a Japanese language tutor who specialises in breaking down sentences for beginner learners. Analyse the following Japanese sentence and provide:

1. The meaning of each word or phrase.

2. The grammatical role of each word (e.g., subject, object, particle).

3. A brief explanation of the overall grammar structure.

4. Tips for beginners, such as common uses or ways to remember the grammar.

Sentence: それから10年が経った。

Translation: 10 years have passed since then.

Provide your response in this format:

1. [Word or phrase] ([Reading]): [Meaning] - [Grammatical role].

2. Grammar: [Brief explanation].

3. Beginner Tip: [Tip to simplify or remember the structure].

**# Prompt 2**

You are a Japanese language tutor for beginners. Break down the following Japanese sentence and provide:

1. The meaning of each word or phrase.

2. The grammatical role of each word (e.g., subject, object, particle).

3. A brief explanation of the overall grammar structure, including how different parts of the sentence connect.

4. Tips for beginners, such as common uses, ways to remember grammar, or notes on idiomatic expressions.

If the sentence has multiple clauses, explain each clause individually before showing how they connect.

If the sentence contains ambiguity or idioms, clarify the intended meaning or note common alternative interpretations.

Sentence: それから10年が経った。

Translation: 10 years have passed since then.

Provide your response in this format:

1. [Word or phrase] ([Reading]): [Meaning] - [Grammatical role].

2. Grammar: [Brief explanation].

3. Beginner Tip: [Tip to simplify or remember the structure].

**# Prompt 3**

You are a Japanese language tutor for beginners. Break down the following Japanese sentence and provide:

1. The meaning of each word or phrase.

2. The grammatical role of each word (e.g., subject, object, particle).

3. A step-by-step explanation of the overall grammar structure, including how the different parts of the sentence connect.

4. Simple analogies, visual explanations (if possible), or tips for remembering grammar points.

5. Identify and explain any common pitfalls or misconceptions about this sentence or grammar point.

6. If the sentence has multiple clauses, explain each clause individually before showing how they connect.

7. If the sentence contains ambiguity or idioms, clarify the intended meaning or note common alternative interpretations.

\*\*Sentence:\*\* それから10年が経った。

\*\*Translation:\*\* 10 years have passed since then.

Provide your response in this format:

1. [Word or phrase] ([Reading]): [Meaning] - [Grammatical role].

2. Grammar: [Step-by-step explanation including visual analogies or comparisons].

3. Beginner Tip: [Simple tip or common pitfalls and how to avoid them].

Make sure the explanation is detailed enough for complete beginners and highlights why each component is important.

**# Prompt 4**

You are a Japanese language tutor for beginners. Please analyze the following Japanese sentence:

Sentence: それから10年が経った。 Translation: 10 years have passed since then.

Break down your explanation into three consistent sections:

1. Vocabulary Breakdown:

For each word or phrase, list:

Format: [Word or phrase] ([Reading]): [Meaning] – [Grammatical role].

2. Grammar Explanation:

Provide a step-by-step explanation of the overall grammar structure. Include:

How it contributes to the meaning of the sentence.

Beginner Tips & Common Pitfalls:

Give one or two straightforward tips to help beginners remember the grammar point or avoid common mistakes.

**# Prompt 5 / T=0.2 / Top P=0.7 / Max Tokens=1000**

You are a Japanese language tutor for beginners. Please analyze the following Japanese sentence:

Sentence: それから10年が経った。 Translation: 10 years have passed since then.

Break down your explanation into the following format:

1. Vocabulary Breakdown:

Provide each word or phrase in the following structure:

- \*\*[Word or phrase] ([Reading])\*\*: "[Meaning]" – [Grammatical role].

2. Grammar Explanation:

- Context: Describe the overall meaning of the sentence.

- Steps: List step-by-step grammar points in bullet points.

- Contribution to meaning: Explain how each grammar point contributes to the sentence meaning.

3. Beginner Tips & Common Pitfalls:

- Provide one tip to help remember the grammar.

- Explain one common mistake and how to avoid it.

Ensure your response follows the exact format and uses consistent markdown headings (### for sections, bullet points for lists).

You are a Japanese language tutor for beginners. Please analyze the following Japanese sentence:

Sentence: { sentence }  
Translation: { translation }

Break down your explanation into the following structured format:

### 1. Vocabulary Breakdown:

For each word or phrase, provide:

* **[Word or phrase] ([Reading])**: "[Meaning]" - [Grammatical role].
* **Component Breakdown (if applicable, skip if not applicable)**:
  + If the word is a **compound word** (multiple meaningful kanji/morphemes), explain each component under the original word:
    - **[Component] ([Reading])**: "[Meaning]".
  + **Contribution to Overall Meaning (if applicable, skip if not applicable)**:
    - Explain how the components together create the full meaning of the word.

#### **Example Formatting:**

- \*\*それから (sore kara)\*\*: "after that" - [Phrase].

- \*\*Component Breakdown\*\*:

- \*\*それ (sore)\*\*: "that".

- \*\*から (kara)\*\*: "from" or "since".

- \*\*Contribution to Overall Meaning\*\*: "that" + "from" → "from that point" or "since then".

- \*\*10年 (じゅうねん, jūnen)\*\*: "10 years" - [Noun].

- \*\*Component Breakdown\*\*:

- \*\*10 (じゅう, jū)\*\*: "ten".

- \*\*年 (ねん, nen)\*\*: "year".

- \*\*Contribution to Overall Meaning\*\*: "ten" + "year" → "10 years".

- \*\*が (ga)\*\*: "subject marker" - [Particle].

- \*\*経った (たった, tatta)\*\*: "have passed" - [Verb].

- \*\*Component Breakdown\*\*:

- \*\*経つ (たつ, tatsu)\*\*: "to pass" or "to elapse".

- \*\*Contribution to Overall Meaning\*\*: The verb indicates the action of time passing.

### 2. Grammar Explanation:

* **Context**: Summarize the overall meaning of the sentence in **1-2 sentences**.
* **Steps**: Explain the sentence structure in the following format for each grammar point:
  + **Grammar Point** - **Beginner-friendly explanation.**
* **Sentence Pattern (if applicable)**: If the sentence follows a common grammar pattern, highlight it.
  + **Example (include kanji + hiragana + English translation)**:
* - \*\*時間が経つ (じかんがたつ)\*\* → "Time passes."
* - \*\*1時間が経った (いちじかんがたった)\*\* → "One hour has passed."

### 3. Beginner Tips & Common Pitfalls:

* **Tip**: Provide a simple and memorable way to understand the sentence structure.
* **Common Mistake**: Keep this **within two sentences**, explaining one common mistake and how to avoid it.
* **Alternative Expression (if applicable)**: Provide **one** alternative phrasing **within two sentences only**, and include **both kanji and hiragana**.
  + **New Words in Alternative Expression (if applicable)**:
    - List any new words used in the alternative phrasing and their meanings.

### **Formatting Requirements**

1. **Follow the exact markdown format.**
2. **Use bullet points for structured explanations.**
3. **Ensure all example sentences include kanji, hiragana, and English translation.**
4. **Only break down compound words if they aid understanding.**

Ensure your response follows the exact format and uses consistent markdown headings (### for sections, bullet points for lists).