Comment Bank Project Documentation

This documentation covers the setup and usage of the Comment Bank project. The project includes functionality to manage subjects, year groups, categories, comments, and report generation using the OpenAI API.

Database Schema

The database consists of the following tables:

1. Subjects
2. YearGroups
3. Categories
4. Comments
5. Prompts

Here is the SQL to create the database schema:

```
DROP DATABASE IF EXISTS comment_bank;

CREATE DATABASE comment_bank;

USE comment_bank;


CREATE TABLE Subjects (

    id INT AUTO_INCREMENT PRIMARY KEY,

    name VARCHAR(255) NOT NULL
```

```
);


CREATE TABLE YearGroups (

    id INT AUTO_INCREMENT PRIMARY KEY,

    name VARCHAR(255) NOT NULL

);


CREATE TABLE Categories (

    id INT AUTO_INCREMENT PRIMARY KEY,

    name VARCHAR(255) NOT NULL,

    subjectId INT,

    yearGroupId INT,

    FOREIGN KEY (subjectId) REFERENCES Subjects(id) ON DELETE CASCADE,

    FOREIGN KEY (yearGroupId) REFERENCES YearGroups(id) ON DELETE CASCADE

);


CREATE TABLE Comments (

    id INT AUTO_INCREMENT PRIMARY KEY,

    text TEXT NOT NULL,

    categoryId INT,

    FOREIGN KEY (categoryId) REFERENCES Categories(id) ON DELETE CASCADE

);


CREATE TABLE Prompts (

    id INT AUTO_INCREMENT PRIMARY KEY,

    subjectId INT,
    yearGroupId INT,

    promptPart TEXT NOT NULL,

    FOREIGN KEY (subjectId) REFERENCES Subjects(id) ON DELETE CASCADE,

    FOREIGN KEY (yearGroupId) REFERENCES YearGroups(id) ON DELETE CASCADE

);
```

Setup Instructions

1. Clone the repository and navigate to the project directory.
2. Install the required dependencies using npm:

   npm install

3. Set up the MariaDB database and import the schema provided above.
4. Update the `server.mjs` file with your OpenAI API key and database credentials.
5. Start the server:

   npm start

API Endpoints

The server exposes the following API endpoints:

1. Subjects
   - GET /api/subjects: Fetch all subjects
   - POST /api/subjects: Create a new subject
   - PUT /api/subjects/:id: Update a subject
   - DELETE /api/subjects/:id: Delete a subject

2. YearGroups

   - GET /api/year-groups: Fetch all year groups

   - POST /api/year-groups: Create a new year group

   - PUT /api/year-groups/:id: Update a year group

   - DELETE /api/year-groups/:id: Delete a year group

3. Categories

   - GET /api/categories-comments: Fetch categories and comments by subject and year group

   - POST /api/categories: Create a new category

   - PUT /api/categories/:id: Update a category

   - DELETE /api/categories/:id: Delete a category

4. Comments

   - GET /api/comments: Fetch comments by category

   - POST /api/comments: Create a new comment

   - PUT /api/comments/:id: Update a comment

   - DELETE /api/comments/:id: Delete a comment

   - POST /api/move-comment: Move a comment to a different category

5. Prompts

   - GET /api/prompts: Fetch all prompts

   - GET /api/prompts/:subjectId/:yearGroupId: Fetch prompt by subject and year group

   - POST /api/prompts: Create a new prompt

   - PUT /api/prompts/:id: Update a prompt

   - DELETE /api/prompts/:id: Delete a prompt

6. Generate Report

   - POST /generate-report: Generate a report for a pupil

7. Import Reports

   - POST /api/import-reports: Import reports and generate categories/comments

Frontend Pages

The project includes the following frontend pages:

1. index.html

   - Allows the user to generate reports for pupils using selected categories and comments.

2. manage_subjects_years.html

   - Provides an interface to manage subjects and year groups.

   - Displays existing subjects and year groups.

   - Allows editing and deleting of subjects and year groups.

   - Allows setting and updating the prompt for each subject-year group combination.

3. manage_categories_comments.html

   - Provides an interface to manage categories and comments.

   - Allows adding, editing, and deleting categories and comments.

   - Allows moving comments between categories.

4. import_reports.html

   - Allows users to import a large number of reports for a particular year group and subject.

- Extracts categories and comments from the imported reports using OpenAI and adds them to the database.