

# CS230[A] – Web Information Processing (2018-19:Semester 2)

Dashboard / CS230[A] (18-19:S2) / Labs / Lab 6-8 / Assignment 4

## Lab 6-8 / Assignment 4

You are required to develop a Hiragana Tutor app, functioning as a single page application, that runs in a web browser. The app should provide functionality for saving and reviewing learner progress using AJAX calls to some accompanying database via a web service you have also created. You may use any learned technology to implement the app, the service, and the database, for example, you may choose to use HTML/CSS/JS (with jQuery) for the tutor app, and use PHP for the service, and MariaDB as the database. *For this assignment, however, is essential that you use AJAX for communication between the app and the database for all persistent data storage.*

Hiragana is one of three character sets (two kana and one kanji) used in the Japanese language. Each Hiragana letter represents a particular phonetic syllable. The concept of a 'letter' has no meaning in Japanese. In modern Japanese, there are 46 basic Hiragana characters. In addition to these 46 basic characters (called gojūon), there are modified of the basic forms to describe more sounds, i.e. 20 dakuon, 5 handakuon, 36 yōon, 1 sokuon and 6 additional characters. English learners of the Japanese typically use a set of words called romaji which is a representation of Japanese sounds using the western, 26-letter alphabet. For example, the hiragana character "あ" has the romaji equivalent "a" (pronounced like the "a" in "car" or the "a" in "awful"). You do not need to be able to speak Japanese for this assignment; you only need to know what Hiragana and its corresponding romaji look like, and how they are structured.

Please note:

- You are required to use the lab computers, and your pre-existing XAMPP installation.
- You can use web resources, i.e. W3Schools.com, books (printed, or electronic), or CS230 notes.
- Communicating, verbally, in writing, or electronically with another student will incur penalty.
- **NOTE: Your code must be able to run on other machines.** If the demonstrator marking it cannot get it run on their machine, you will receive no marks. It is your responsibility to test your code and make sure it works outside of your environment.

**Before submitting your assignment students should check that their solution works in Chrome and Firefox.**

**All work must be submitted via Moodle. Work submitted via other means will not be accepted unless you have prior arrangements with the head demonstrator (Shane McGarry). This code must be submitted as 3 separate .gz files as per the instructions (see #5).**

1. Present a large (or full-screen) display showing the full Hiragana set, black lettering on a white background, including each character's associated romaji, and surrounded by a black box with a drop-shadow. Clicking an individual character should play an audio pronunciation for that kana element. Hovering the mouse over an individual element should show the stroke drawing order for that character. You can find free-to-use stroke order images (as animated GIFs) [here](#). You can find free-to-use audio files from a working link [here](#).
2. Present functionality that allows the user to select a training set of elements for study. The user should then be presented with a quiz, testing recognition of the elements in the training set. The learner should be quizzed on recognition of individual kana elements and the corresponding romaji equivalent, and vice versa. Learners should be presented with four choices, and be able to select one correct answer for each question. Learners should be presented with 10 questions in each quiz and be provided with a percentage score at the end of each quiz.
3. You are required to provide each learner with an option to save their quiz performance for each Hiragana character. If this option is selected, you should authenticate the user, or ask them to create a user/password which will be used for authentication. Once authenticated, saved learner performance should be stored on the server after each quiz, etc. You should use AJAX for all communication with the web service responsible for management of the accompanying learner database. Users may opt to learn without saving, and in this instance all session information is lost once the app closes. Once authenticated, the learner's performance, etc. should be loaded from the server and applied to the current app state.
4. **NOTE:** You must visibly credit all resources (images, sounds, APIs, etc.) used clearly, together with an acknowledgement and link to each resource, in the app. Failure to clearly credit all resources will result in a penalty (marks reduction).
5. **NOTE:** You should submit three primary files for correction:

- a. all front-end files, including HTML, CSS files, JS scripts and libraries, images, sound files, etc. should be zipped into a file called front-end.gz
- b. the web service should be zipped into a file called web-service.gz
- c. the database tables including working data, should be dumped (use command mysqldump to do this) into a file called database.sql, and zipped into a file called database.gz

These three files ( front-end.gz, web-service.gz & database.gz) should be submitted separately. Failure to submit you work correctly may result in a penalty (marks reduction).

6. **Extra Credit:** For additional credit, implement any, or all of, the following functionality:

- a. This user should be allowed to save and delete training set information. All saved progress should, therefore, also incorporate progress for training sets in addition to individual kana elements (*2 points*).
- b. The Hiragana chart display should incorporate a button, which toggles learning progress, by way of colour-coding the individual hiragana characters in the main display. Characters that are known should be presented on a black background with white lettering. You can use shades of grey to indicate progress on characters not yet known. Progression information should be automatically updated upon authentication (*2 points*).
- c. Learners should be presented with timed recognition quizzes in addition to the regular quiz. Timed performance also needs to be saved, if you are including timed quizzes in your app (*2 points*)
- d. Your app should present the advanced learner with additional Hiragana training (i.e. 20 dakuon, 5 handakuon, 36 yōon, 1 sokuon and 6 additional characters). If you research hiragana table representations you will find the best way to extended to display in order to include the additional elements (*2 points*).
- e. Your app should present the advanced learner with legitimate two-character Hiragana / Romaji timed quizzes, for example, "おい" (oi) oil/nephew (*2 points*).

Last modified: Wednesday, 13 March 2019, 12:00 PM

---

◀ Lab 5 / Assignment 3

Jump to...

Assignment 1 ▶

You are logged in as STEWART THOMAS HETHERINGTON ([Log out](#))

[Data retention summary](#).