GSoC'16 Proposal – Enhancement and Offline Support for Wikitolearn Editor

About the Proposal

The idea is to improve the current Wikitolearn VisualEditor by adding

- Offline Support
- Autocomplete feature
- Syntax Completion feature
- Snippet Feature

Background (About VisualEditor)

VisualEditor is a project to provide a visual online rich-text editor as a MediaWiki extension to Wikipedia. It was developed by the Wikimedia Foundation. In July 2013 the beta was enabled by default, with the ability to opt-out, for Mediawiki.org and several of the largest Wikipedias. VisualEditor is a Javascript contenteditable HTML+RDFa editor.This visual editor will never replace the classic wikitext editor, but it is simple to use and preferred by new editors. Wikitolearn, is currently using the old wikieditor and as said, will be switching over to VisualEditor very soon. But, why VisualEditor?

- VisualEditor is highly modular.
- Replace/Extend existing visualeditor tools.
- Support for more than 300+ languages
- Wikitext takes time to learn for a newbie.

Currently, Visual Editor lack features like offline editing. Suppose, a user is editing a page, and suddenly the network goes down, then the work done by the user also gets disappeared.

The Project

1. **Offline Support :**

    a. ***Overview***: Basically, I will be building a Chrome Browser plugin (with the help of HTML5,CSS3 and JavaScript), which would cache the web page in which editing is taking place, ask the user to edit it offline, save it to the local storage and as soon as the network permits, push the changes.

    b. ***Capabilities of the plugin*** :

        i. Ability to cache the web page

        ii. Ability to save and sync the changes: I will use the **chrome.storage API** to store, retrieve, and track changes to user data. To store data, I will be using **storage.local** which would automatically store data locally. When the network is down, it would store data locally, and the next time,

network's up, it syncs the data.

iii. Ability to detect changes in internet connectivity: I will use the **navigator.online** property that returns *true or false* depending on whether or not the plugin has network connectivity.

c. ***Difficulties to overcome:***

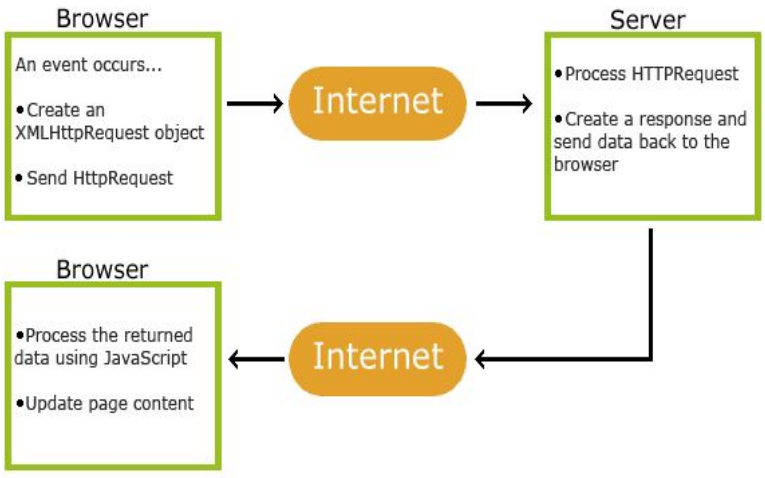
i. Upload Media doesn't work if the network goes down.
ii. I would also add this user case: the user goes offline, edits the page and once it come back online, the page has been edited by someone else.

2. **Autocomplete feature :**

a. ***Overview:*** I will use AJAX for linking awesomplete (https://github.com/LeaVerou/awesomplete) to the Visual Editor for autocomplete feature. Awesomplete is an ultra lightweight, customizable, simple autocomplete widget with zero dependencies, built with modern standards for modern browsers.

b. Working: I will be storing all the dictionary terms in a db file and then when the user starts typing a new word after a space, I will use an AJAX query which would reply to user a list of words matching. Now, I have the words in the Javascript response, I will put that words in a *for loop*, and print them line by line below the cursor that would allow the user to select the correct word and then autocomplete is done.

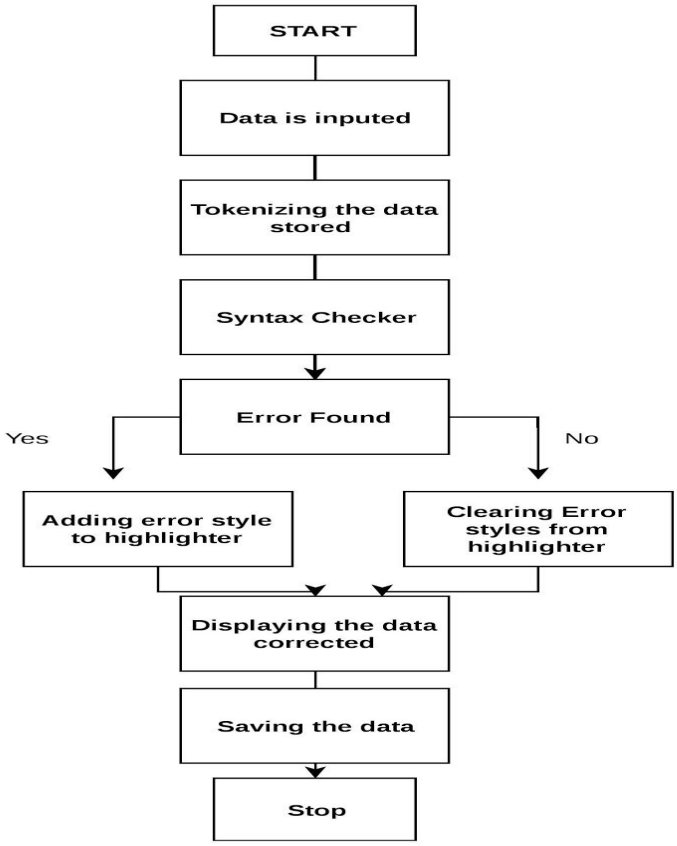
This's how an AJAX query works :



3. **Syntax Completion Feature :**

a. ***Overview*** **:** I would be making a visualeditor or mediawiki extension capable of reorganizing code formatting, syntax highlighting, including indentation.

b. ***Working*** **:** The following flowchart would explain how this extension would be working:



4. **Code Snippet Feature :**

a. ***Overview :*** Code snippets are small blocks of reusable code that can be inserted in a code file using a context menu command or a combination of keys. Adding snippet feature in the editor will allow users to create a basic template structure.

b. ***Implementation*** **:**This would be working quite similarly as the autocomplete feature i.e. whenever a user types chevrons (angle brackets '<'), I will use an AJAX query which would reply the user, a list of tags matching. Now, I would have the codes in the Javascript response and as soon as the user clicks on the specific tag, it would print the code. There are already mediawiki gadgets for visual editor that gives the possibility to implement snippets.

Timeline

Apr 23rd – May 22nd [30days]

An in-depth check of codes of VisualEditor will be the first thing I will do. During this period, I would like to get in touch with my fellow peers who have been approved for GSoC. Get in touch with senior developers, I will make sure that they know me and know what idea I'm working on. Also important to get in touch with Developers who are working on enhancing VisualEditor. Will read necessary documentation on VisualEditor, write small codes and participate in the organization's community.
Most important for me, I will try to lay out a foundation, so that when the official coding begins I will be able to give my project kick start.

May 23 Officially coding begins for **Google Summer of Code, 2016.**

May 23rd – June 20th [28 days][8+ hrs per day][40+ hrs per week]

During this period, I'll try to completely implement autocomplete feature,syntax completion feature as well as the snippet feature. Also, I'll start working towards the offline feature, which would take a longer time to complete.

June 20th - June 27th [7 days][8+ hrs per day][40+ hrs per week]

Mid-term Evaluations

June 27th - August 10th [44 days][8+ hrs per day][40+ hrs per week]

During this period, I'll be working on implementing the offline support plugin for visual editor.

Aug 11th – Aug 23rd [12 days][8+ hrs per day][40+ hrs per week]

This is the Firm 'pencils down period. I will test and review all the codes i have written by that time.

About Me

I'm a 1st year Computer Science student in Amrita School Of Engineering, Amrita University, Amritapuri Campus, India. Apart from academics, I have great interest in Free & Open Source Software. Also, I'm an active FOSS@Amrita Club member.

When I got to know about Gsoc, a desire inside me burned to take part in it. One of my seniors, Mr. Harish Navnit, being himself a KDE contributor, asked me go through the KDE project ideas, and I found this project really fascinating.

I'm well-versed with HTML, CSS, JS, PHP & Bootstrap. Also, I'm a rookie in C and Python. I'm always keen to learn new things.

Location (City, Country and/or Time Zone): Kollam, India, UTC+5:30

References

https://www.mediawiki.org/wiki/Extension:VisualEditor

https://www.mediawiki.org/wiki/Extension:Gadgets