UTRRS Documentation

Project Purpose:

The current UTRRS(Unicode Text Rendering Reference System) site should have new User Interface along with some new features. And the comparison part should also be automated which has been done by now manually.

Level of Functionality:

- **1st Level of functionality:**

  Navigation Bar

  When user opens the home page of the website, the user get a new User Interface with new navigation bar containing:

  - Home – User can click on it anytime to directly go to the home page of the website.
  - About – User can know about UTRRS by click on it.
  - Check Fonts – User can check whether the fonts used in UTRRS are installed in his system or not. Here the User can see the commands to run according to different languages, to check whether the following fonts are installed in his system or not.
  - Feedback – User can click on it, to directly send his feedback about anything to the admins of UTRRS.

  Sidebar

  At the left side of the home page, User get a list of different languages where the User can get to know about different locales by clicking on any particular one.

  Footer

  At the bottom of the home page, User can see another list of options containing:

  - History – User can click on it to know about the history of UTRRS.
  - RoadMaps – User can click on it to know about the future road maps of UTRRS.
  - Development – User can click on it to know about the main development of UTRRS in fuel page.
  - Contact Us – User can click on it, if he wants to contact the admin of UTRRS.

- **2nd Level of functionality:**

  When the User click on any particular language, he will be redirected to that language. In the home page of that language user get all the information regarding that particular language like brief about that language, codepoint, reference image, the character itself and its description.

On the top of that page user get three option containing:

- Codepoint – User can click on it to go directly to the codepoint page of that language. This page contains only the codepoint of that particular language.
- GSUB – User can click on it to go directly to the glyph substitution page of that language. This page contains only the glyph substitution of that particular language.
- GPOS - User can click on it to go directly to the glyph positioning page of that language. This page contains only the glyph positioning of that particular language.

On the left side of the page User still get that language list from where he can directly switch to other languages.

On the top of information table of language, User get an option of Search where he can search anything to filter his required data. User can put codepoint, character, part of codepoint, description or part of description in the search bar to search his required data. Along with Search, User get an option to see the table in different size of entries per page like 10, 20 etc. User can choose different option from the list to satisfy his requirements.

At the bottom of the table, User get the information about the total entries the table. And pagination for that table, by using pagination User can jump to any particular page he wants to.

- **3rd Level of functionality:**

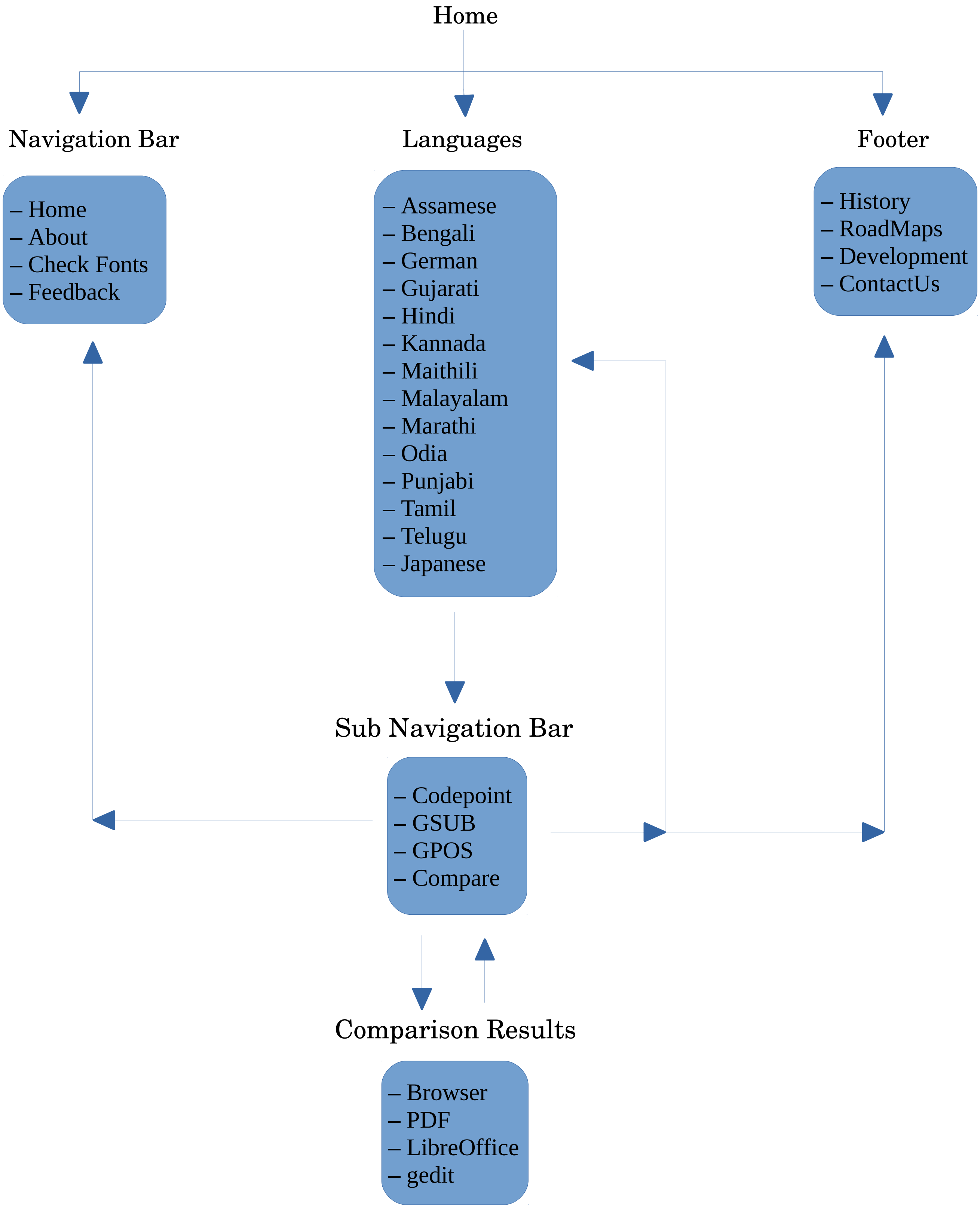
When User clicks on any language, in the top navigation bar a new option will appear named “Comapre”.

By clicking on this Compare option, the User can compare the character and its reference image which had been done so far manually. As each locales have different numbers of characters, the comparing part will take time according to it. After the comparison finished, User get an statistics of matched and unmatched characters along with some options containing:

- Browser – User can click on it to see which character is matched or unmatched and how much they are matched in percentage.
- PDF – User can click on it to see the results in PDF format. When user clicks on it, a PDF will be automatically downloaded containing the browser results.
- Libre Office - User can click on it to see the results in CSV format. When user clicks on it, a CSV will be automatically downloaded containing the browser results.
- gedit - User can click on it to see the results in TEXT format. When user clicks on it, a TEXT will be automatically downloaded containing the browser results.

User can click on the upper cross sign, Close button or click outside the pop up window to close the comparison window and get back to that particular language.

Work flow Diagram:



Data flow Diagram:

- Home → (home.html) Takes data from html page and shows in Browser.
- About → (about.html) Takes data from html page and shows in Browser.
- Check Font → (check_font.html) Takes data from html page and shows in Browser.
- Feedback → Mail window pop up.

- History → (history.html) Takes data from html page and shows in Browser.
- RoadMaps → (roadmap.html) Takes data from html page and shows in Browser.
- Development → Redirect to fuel page.
- ContactUs → Mail window pop up.

- Languages:

  (Assamese) → Assamese homepage (assamese.html) → (def assamese()) Assamese function from view.py → The request is not ajax so execute the else part of function and render the assamese.html page.
  Take data from path : static/lang/as_IN/font/data/master_as.txt

  Assamese homepage (assamese.html) → Codepoint → Codepoint page (as_codepoint.html) → (def assamese_codepoint()) Assamese codepoint from view.py → The function executes and render the as_codepoint.html page.
  Take data from path : static/lang/as_IN/font/data/codepoint/master_as_IN.txt

  Assamese homepage (assamese.html) → GSUB → glyph substitution page (as_gsub.html) → (def assamese_gsub()) Assamese gsub from view.py → The function executes and render the as_gsub.html page.
  Take data from path : static/lang/as_IN/font/data/gsub/master_gsub_as_IN.txt

  Assamese homepage (assamese.html) → GSUB → glyph positioning page (as_gpos.html) → (def assamese_gpos()) Assamese gpos from view.py → The function executes and render the as_gpos.html page.
  Take data from path : static/lang/as_IN/font/data/gpos/master_gpos_as_IN.txt

  Compare → (def assamese()) Assamese function from view.py → (Ajax Call) As the request is ajax if part of the function executes → Take the characters, see their images from (static/lang/as_IN/font/tmp_images) and compare those images with the reference images (static/lang/as_IN/font/images/) and gives the results in different format like pdf, csv and text file. All these are created during the ajax call for comparison.
  Take data from path : static/lang/as_IN/font/data/master_as.txt

PDF → Download PDF file → (def assamese_pdf()) Execute this function for downloading the file.
File path: static/lang/as_IN/font/tmp_images/assamese-report.pdf

Libre Office → Download CSV file → (def assamese_csv()) Execute this function for downloading the file.
File path: static/lang/as_IN/font/tmp_images/assamese-report.csv

gedit → Download TEXT file → (def assamese_txt()) Execute this function for downloading the file.
File path: static/lang/as_IN/font/tmp_images/assamese-report.txt

- Others languages:
  Procedure, file, path all are same as above Assamese language. User just have to change the path, function and file with proper name of that particular language.
  For eg. for Bengali, it will be → bengali.html, bn_codepoint.html, bn_gsub.html, bn_gpos.html etc

Modules Used:

- django
- os
- filecmp
- csv
- collections
- FileWrapper
- reportlab
- terminaltables
- itertools
- PIL

Prerequisites:

- Django==1.10.4
- Pillow==3.4.2
- reportlab==3.3.0
- terminaltables==3.1.0

Deployment Procedure:

Example to deploy in openshift: https://github.com/openshift/django-example

Maintenance Procedure:

If in future some new characters or language are being added with the current languages. User should create the reference images with “hb-view” which is a part of Harfbuzz rendering engine. As well as user should add the character’s codepoint, image path, description etc. in the master text file so that it will be available to render and compare.

“hb-view” allows user to easily create .svg as well as .png images. .svg images are used in this project to enable flexibility when the page is zoom on or zoom out. .png images are used in comparison part.

User should change the view.py as well as others files if new languages are added in future.