IndicWiki Internship Documentation

-Tourist Destinations

—————————————————————

**Domain**:

The domain we worked on was ‘***Tourist Destination***’, where our aim was to generate Telugu Wikipedia articles for 20000+ destination places containing as much information as possible about it.

**Team Members:**

|  |  |
| --- | --- |
| **Member Name** | **Email-Id** |
| Sreeja Dacharla | sreeja.d20@iiits.in |
| Rahul Kanna | kannarahul05@gmail.com |

**Phase1: Data Collection**

**Sources/ Sites**

We browsed and searched for many websites where we could get the appropriate data. Many of the websites had the information but the data wasn’t structured. At last, we found some good sources/sites where we were able to scrape data.

The following are the websites we used for data collection.

i) <https://www.google.com/travel/hotels>

ii) [https://en.wikipedia.org/wiki/](https://en.wikipedia.org/wiki/Main_Page)

We were able to collect the maximum amount of data from these websites. We had to manually collect data for some attributes like temperatures, best time to visit etc.

Here is the list of the attributes we were able to collect from the websites listed above.

1. Name
2. City
3. State
4. Country
5. Continent
6. Address
7. Longitude
8. Latitude
9. Images
10. Wiki links
11. Destination type
12. How to reach
13. Hotels nearby
14. Best months to visit
15. Temperatures
16. Ratings

**Tools Used:**

1. **BeautifulSoup**: It is a python library used for scraping data from web pages using parsers like html5lib, lxml etc. It sits atop an HTML or XML parser and provides Pythonic idioms for iterating, searching, and modifying the parse tree. More than 90% of our data collection work was performed with the help of this tool. This was very handy as we were dealing with static web pages.
2. **requests**: Requests is a Python library built to handle HTTP requests in python easily. It allows you to make GET, POST, PUT and other types of requests and process the received response in a flexible Pythonic way. With the help of this library, we request the website and parse it into a parse tree with the help of BeautifulSoup.
3. **selenium**: Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts. Selenium was a bit difficult to work with as compared to BeautifulSoup. But still, it's the best choice to work when we are dealing with dynamic web pages.

**Image Scraping:**

Image scraping is done by us using SPARQL and manually searching for the places for each country and tourist destination type.The SPARQL code we used for Image scraping is given in the link below.

[Github Link](https://github.com/sreeja2208/Indicwiki/blob/2f9c39b85487f67480504e0f1dd6a656c8d27614/Imagescrapping.txt)

**Data Storing:**

We stored our entire data in excel sheets and google sheets because it is easy to use and easy to work with.

**Phase2: Data Cleaning**

During the data collection part itself we handled this situation by storing null/none when an attribute was missing for a destination place. This helped us a lot by reducing the work of data cleaning. Most of the data we collected was consistent except for the attribute description where we had to manually remove the special characters and destination type null/none values where we had to search them manually.

**Phase3: Version Control**

GIT is the version control tool we used to store and share the data among ourselves. Pushing the code into GITHUB and pulling the file by another member from there and pushing it back to the repository with updated changes was really helpful rather than sharing a copy each time when there are any updates in it.

**Phase4: Translation and Transliteration**

We easily translated the data as it was consistent. The data we collected was atomic values which got perfectly translated into Telugu. Hence, there wasn’t much work in transliterating the data except for the description attribute.

**Tools Used:**

1. **Google Translate**: It is an online translating tool. There’s a python library called GoogleTrans which uses this Google Translate via API calls. Most of the translation part was done with the help of this tool.
2. **Bing Translator:** It is also an online translating tool powered by Microsoft. The transliteration part was performed using this tool as it converted a sentence into a meaningful Telugu sentence.

**Phase5: Sample Article**

We have written a sample article with all the available attributes. This article gave us an abstract on what our future articles will look like.

[Link to article](https://docs.google.com/document/d/1fSwqYH5IywDAJWG4ypcvQvJP8-52QIRagdQCseE4OnI/edit?usp=sharing)

**Phase6: Jinja Template**

Tools Used:

1. **Jinja2**: It is a templating language. Jinja2 is a python library mainly used to create generic templates. With the help of Jinja2, we can create HTML, XML etc pages that are rendered to the users via an HTTP request.

[GitHub Link](https://github.com/RahulKanna05/IndicWiki-Tourist-Destination/blob/main/template/template.j2)

We made sure all the edge cases are handled perfectly without disturbing the structure and meaning of the sentences. We also added randomisation to each sentence so that every article looks different and doesn’t have the same sentences.

**Phase7: XML Generation**

Based on the sample article we created, we generated the xml file which consists of all the destination places we worked gathering data on. We used a recursive code to write the code for us when provided with the data excel sheet.

The sample xml: [GitHub link](https://github.com/sreeja2208/Indicwiki/blob/0ab198435ef18ff238ba66660002d5e99df2dbae/sample2.xml)

The code link: [ipynb link](https://colab.research.google.com/drive/1Onn_CSGL1kmaTYkFrCHIs5L8MADOqLUT?usp=sharing)

The final xml link: [xml file link](https://drive.google.com/file/d/1WB-zcZfTNvyWo5mP3QxsO2UNm3ks2-ND/view?usp=sharing)