Subteam 10-2 Assignment 2 Report

Fucheng zhuang,Peng Du

Section I: Summary

For this project, we decided to do the frontend and backend separately, and connect them using the REST API.

As the subteam that is primarily responsible for the backend, we have decided to build a functional backend API using python and Flask.

There are lots of options when deciding which programming language to use, but we have decided to use **python** for the following reasons:

1. **Ease of use:**

   Python is a general purpose high-level language that is considerably the easiest to learn and use. According to statistics, python will be known by 48.07% of programmers by the end of 2022, which is much higher than other languages like Java(33.27%) and C++(22.55%) (Vailshery). Therefore, using python makes the project easier for others to develop and maintain.
2. **Maturity**

   With no doubt, Versatile and Mature Programming Language
3. **Domain covered by python**
   - Machine learning / Artificial intelligence
   - Desktop GUI
   - Data analytics and data visualization
   - Web development
   - Game development
   - Mobile app development
   - Embedded systems

https://somenplus.blogspot.com/2020/11/top-7-domains-to-expertise-after.html

4. **Abundance of libraries:**

   Since our project is a search engine, the backend logic would involve some complicated concepts, such as spell-correcting and content-searching, which are difficult to be hand-coded from scratch. Python has a vast collection of libraries that can be used for various purposes, which are really helpful in backend development.
5. **Wide support for frameworks:**

   Python supports various modern web development frameworks, such as Flask and Django. In addition, thanks to the large developer community of python, there are third-party packages to support frameworks and platforms that originally don't support python(e.g. airtable).

Originally, we decided to use Django since we learned Django in CSC309. However since our partner has already implemented the entire database using airtable, we did some research, decided to try - **Flask** for the following reasons:

1. **Database has been implemented by our partner**

   We don’t need to create models, the database is already implemented, Django is great for Model View Controller(MVC) and build in ORM which we don’t really need
2. **Ease to use**

   Flask is well-integrated with python and requires minimum boilerplate code, which allows us to focus more on the searching logic, instead of struggling with the code templates and syntax issues. And if someone wants to take over our project in the future it’s easier and faster to understand the code

3. **Minimum complexity**

   It’s easy for deployment and testing. First stage’s API testing using postman is sufficient.

Section II: Individual Contribution

- Peng Du:
  1. Created a python script to parse the data from our partner’s AirTable Database, and create hashmaps for quicker accesses.
  2. Designed backend API endpoints and created corresponding documentations.
  3. Created the Flask application in python and implemented 9 API endpoints.
  4. Developed edit distance algorithm, spelling autocorrection and the API endpoints for keyword association, retrievement and searching results.
  5. Write the summary for the options and decisions in the subteam report.
- Fucheng Zhuang:
  1. Also write the subteam document.
  2. Communicating with the subgroup that is responsible for UI, make a few improvements on the UI, also connect with the subgroup that is for testing.
  3. Add readme instructions in Github.
  4. Create the basic UI that connects with Flask to perform the task.
  5. Deploy to the AWS.

Section III: User Instructions(For TA testing)

Step 1) Check out the document of raw json data given by the partner, they used airtable to implement the database.
airtable output

Step 2) Go to http://3.21.168.194/

Step 3) Try the first search feature，it also has spelling auto correction feature
Example：Try “kevin”, the output is the record ID, name + company
Step 4) Try the second search feature, use record ID to search the detailed information. This feature will be combined into the first feature in the future.

Step 5) Try the Third search feature, use year to search, so far it has to strictly follow the format like inside quotations ("2019–20"). In the future this will be integrated into a filter feature.

Step 6) We came up with 9 APIs, all of them are implemented，but doesn’t have an easy UI for demo so far, you can try to add the Endpoint to http://3.21.168.194/ to test out other features if you like.
CSC301 API Documentation

Work Cited

Vailshery, Lionel Sujay. “Most used languages among software developers globally 2022.” *Statista*, 9 August 2022, https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/. Accessed 16 February 2023.