KENNESAW STATE UNIVERSITY

IT Alumni Database ETL System

(Placed 3rd at C-Day for Undergraduate Capstone Projects)

Project website:

<https://sites.google.com/view/it-alumni-database/home>

Course: IT 4983

Members:

Ricky Parks (Team Lead)

Zack Downing

Vy Doung

Desiree Smokes

Report Date: 4/29/2018

Table of Contents

Executive Summary1

Background2

Project Outcomes and Achievements Summary 3

Project Planning and Management Summary 4

Team Reflection 5

Appendix6

1. **Executive Summary**

The Information Technology Department within the College of Computing and Software Engineering of Kennesaw State University is in need of a more efficient method of procuring and storing current alumni information. The method that exists currently is tedious and involves faculty individually entering alumni information into a database manually. This project seeks to automate the process and ensure that no time is wasted in the collection of the desired data. A web crawler was constructed by Hang Yu (MSIT student) to search and scrape data from LinkedIn profiles and export the information to text files. Our team converted the output data from the web crawler to a JSON format that we then parsed using a Python script. This script took the JSON data, organized it, and inserted it into a SQL Server database that we created for this project. The Python script is capable of inserting and updating student data and we avoid duplicate entries by using the unique LinkedIn profile ID as the primary key in the SQL Server database. This solution fulfills the main aspects of the assignment as laid out in the project charter by first automatically performing ETL on alumni data from LinkedIn and secondly automatically performing updates on existing alumni information inside of a database.

1. **Background**
   1. Project Background: The major goal of this project was to see if it was possible to change the way that Kennesaw State University and potentially other Universities keep up with their students once they graduate. Currently, for the KSU IT department to track graduates, an employee must manually visit LinkedIn profiles and enter information. This is a very tedious task. Our objective was to automate this process. The major problem we knew we were going to face ahead of time was the task of extracting and transforming the data into a useful format that can be stored into a database.
   2. Scope:
      1. Develop a script capable of crawling LinkedIn and extracting the data.
      2. Develop a database capable of storing information about alumni.
      3. Develop scripts that can take the information extracted from LinkedIn, transform it, and store it into the database.
   3. Deliverables:
      1. Design and develop a database
         1. Choose the environment
         2. Install the server
         3. Complete the conceptual design
         4. Complete the logical design
         5. Create the Schema
         6. Execute the DDL on the server
      2. Insertion Script
         1. Connect to the database using Python
         2. Collect and parse the data
         3. Logic to check if record exists in the database
         4. Logic to insert new records
      3. Update Script
         1. Check which items need updating
         2. Work on logic that will update records automatically
         3. Use the web crawler to update records
   4. Technical Background: To complete this project, there were many different tools and technologies we needed. First, we needed a scripting environment. For this, we used the Python Environment paired with PyCharm Community Edition. PyCharm community edition is an Integrated development environment used by Python developers to make writing Python scripts easier. Next, we needed a database environment. For this, we chose a relational database using the SQL Server management studio environment. For us all to work on and access the database, we hosted the database using Microsoft Azure cloud services. Next, we needed a technology to transform the data into a format that will be easy to read from. For this, JSON was used. A few Python libraries to tie everything together were needed as well. The BeautifulSoup (Bs4) library was used to crawl data from LinkedIn search results and profiles. The Requests library was used to create logon sessions on LinkedIn. Lastly, the PyODBC library was used to connect to the database.
2. **Project Outcomes and Achievements Summary**
   1. General Outcome: Overall, the project is considered a success. We successfully extracted information from LinkedIn, transformed the data, and loaded it into a relational database.
   2. Database Deliverable Outcome: Using SQL, we designed and built a database that could store information pulled from LinkedIn.
   3. Insertion Deliverable Outcome: We managed to open the JSON files generated from the web crawler, read the data into data structures for each alumnus, and insert their information into the database. One minor problem we discovered was the possibility of making too many connection requests at one time. This could potentially get the account you’re using suspended. Another issue we ran into is the problem of trying to view LinkedIn profiles of people that aren’t considered “in your network” LinkedIn doesn’t allow basic-tiered accounts to view “out of network” profiles. However, this can be remedied by purchasing a higher-tiered account.
   4. Update Deliverable Outcome: The update script allows the user to update records that already exist in the database. It is like the insert script. However, most of the logic checks to make sure the information that is crawled doesn’t already exist in the database. This is to ensure duplicate data isn’t being inserted. If an alumnus makes changes to their account, the database is updated after a user executes the update script.
   5. Technical Summary of Solutions:
      1. Technical solution: For our solution, we used a combination of reading from and writing to files. In short, the web crawler first connects to the search results page of LinkedIn using the user provided search string. It then retrieves the unique LinkedIn profile links and writes them to a file. Next, the crawler reads from that file, visits each profile link and crawls the data. Then the crawler parses the data and creates a new JSON file for each student. The Insert and update scripts read these JSON files to insert/update data in the database.
      2. Analysis and design summary: Using the requirements list given to us by Dr. Li, we constructed an ER diagram that represented how the data was going to be stored in the database. A few changes to this design were made over the course of the project due to Dr. Li and everyone deciding on a simpler way to store the data. We originally had a few extra tables that weren’t being used such as the ‘Address’ table that we decided needed to be removed from the design. As for the scripts themselves, we used several flowcharts to map out the flow of data from LinkedIn to the database.
      3. Technical Features: The primary features of our solution are twofold. First, the user can insert data by using one of two methods. The user can insert data by searching for a specific student by entering their name, field of study, school, and graduation year. This method uses a file that the user can edit by entering student info on separate lines. The user can also choose to run a different method that will try to retrieve information about multiple students by providing the same information as the other method, minus the student’s name.
      4. Documentation Summary: Our documentation mostly consists of flowcharts and ER diagrams. Our test plan was to test the scripts separately first. For example, we made sure the crawler, insert, and update scripts worked individually before trying to get them all to work together.
      5. Other Documents: We included a manual that will inform users how to use and install the scripts. We also included the database schema, so a user can create a local database instead of using the one that is in the cloud now.
3. **Project Planning and Management Summary**
   1. Overview: For management of the project, the plan was to split up into Database/Design and Python implementation teams. However, in the end, we all had to contribute to multiple different parts of the project. The general technique was to have two people to primarily work on the Python code and two people to update the database and design aspects whenever changes had to be made.
   2. Project Process:
      1. **Milestone 1**: The objective of the first milestone was to get a working database. This mostly involved creating the conceptual and logical design. Vy and Desiree contributed a lot to this stage. They had a preliminary ER diagram work with given to us by Dr. Li. They made changes to it based on what me and Zack discovered while trying to review the Python we needed to know.
      2. **Milestone 2:** The objective of the second milestone was to be able to read the crawled data and insert it into the database. The major challenge here was to decide on an appropriate format. We all put our heads together and decided that JSON would work better than excel.
      3. **Milestone 3:** The last milestone’s challenge was to be able to update data that already existed in the database. The major problem here was to figure out how to deal with duplicate data. Each of us participated in multiple brainstorming sessions to finally figure out how to tackle this problem.
   3. Major Contributions:
      1. Zack Downing: Website, Update script deliverable
      2. Ricky Parks: Python script for updating, inserting, and modifying web crawler
      3. Vy Duong: SQL design and help with python scripting
      4. Desiree Smokes: SQL design, flowcharting, and python scripting
   4. Workload Summary Hours

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Milestone 1** |  | **Milestone 2** |  | **Milestone 3** |  |
| Member | Hours | Member | Hours | Member | Hours |
| Ricky | 26 | Ricky | 30 | Ricky | 39 |
| Zack | 25 | Zack | 26 | Zack | 33 |
| Vy | 27 | Vy | 22 | Vy | 35 |
| Desiree | 24 | Desiree | 22 | Desiree | 34 |

1. **Team Reflection**
   1. Project success factors: The project’s success factors were to get a working database that’s capable of storing data from LinkedIn. The project owner, Dr. Li was satisfied by our ability to transform the information pulled from the site, and converting the data into an acceptable form.
   2. Team collaboration experiences
      1. We collaborated on this project via group messages. This didn’t always work out because we were unable to see each other’s screens.
      2. For meetings, we met using Discord or at the Marietta J building with Dr. Li. We mostly met towards the end of each week. This was fine for the most part, but it would better if we had more defined meeting times. Also, notifications via discord didn’t always appear right away to us.
   3. The biggest technical challenges we faced was converting the information that was pulled from LinkedIn to a format that the database would accept. For example, we had a tough time converting the graduation dates and job end dates to an acceptable format. We tackled this challenge by splitting the years, months, and days up into an array. Then, depending on whether the day of the month was a single digit or not, we appended an “-01” or “-“to the string.
   4. Areas to improve: The biggest area to improve are our communication practices. Dr. Li commented on this a few times about us not having a set day/time of the week to meet.
2. **Appendix**
   1. Project Files:
      1. There is a folder titled “scripts + installation instructions”. This folder contains the scripts for the project. Included is a readme file that informs users how to install and use the scripts. It includes instructions about topics such as which functions to run and how to setup the database.
      2. ER Diagram – This is the diagram of the database
      3. DDL.txt – This contains the data definition language. We included this file, so someone can use it to create a local database on their computer.
   2. Progress Report Files: This folder includes the three milestone reports, the project plan, the gantt chart, and the logs.