**Title of the project: JobFinder**

**Group members:**

Brogan Murray: murraybm24@uww.edu

Corban Larson: larsonci07@uww.edu

Jake Cramer: cramerJ04@uww.edu

1. **Motivation and project description**

This program serves as a job listing bulletin board–providing the job details, website, company information, and contact information. It allows users to create an account, save listings, add/delete/modify listings, and search for specific listings. The program's main priority will be the search function, which will implement custom or predetermined search filters to tune results. Examples of the search function would be location, salary range, education level, and company size/name. We will construct a C++ standalone application for our project using a preexisting tool such as QT. A database is needed for a diverse, updated, and abundant listing application. The following DBMS characteristics are essential for the program's functionality: user concurrency, query language, reporting tools, and efficient lookups.

1. **Datasets (**[**Link**](https://www.kaggle.com/datasets/ravindrasinghrana/job-description-dataset)**)**

The *Job Dataset* by Ravender Rana offers a vast artificially-created dataset simulating accurate job listings. We chose this dataset due to its 1,000,000+ entries, each with a unique jobID to identify the listing. This job listing approach to data entry met the requirements for hosting a job searching application. Additionally, including twenty-three columns allows for leeway when creating a tailored application. Each job in the dataset has an ID, title, salary, education requirement, skills, benefits, etc. These descriptive attributes make for a detailed, engaging experience for the application user. Along with a job description, there is also a host site and contact information for the company offering the position. The host site and company offer additional vectors for searching for relevant job opportunities. The most significant advantage to utilizing this dataset is that entering specific values (ex., $50K starting salary) would narrow down the listings while still offering a wide selection.

Compared to many of the databases we considered, the *Job Dataset* avoids the pitfalls of storing a large amount of nullable values. Since this set represents realistic job listings, there aren’t null inputs, allowing conciseness. It reduces time on narrowing columns of unusable data and provides a more fortified job listing, further improving the user experience.

1. **Project schedule**

| **Week** | **Description of expected deliverables** |
| --- | --- |
| 1 | ER diagram finished |
| 2 | Translation of ER diagram to relational. Normalization |
| 3 | Table and relationship designs |
| 4 | Populate tables, setting up keys, constraints etc.. |
| 5-7 | SQL |
| 8 | GUI |
| 9 | Evaluation |
| 9, 10 | Report writing |

1. **Assessment plan**

Being a user-centered design, data integrity, and availability are of the utmost importance. The job bulletin must successfully store user information, allow for modification/access/deletion of job listings, and allow a user to save listings. To test our program’s functionality, we must complete three successful attempts of the following functions to ensure they’re correctly functioning:

* Create a user account - allows users to register accounts with the application. It prompts the application to store user information for future use.
* Save listing - allows users to save listings into a favorites list for future reference. Any saved listing must show under the favorites list unless the user deletes it.
* Access/Modify/Delete Listing - Users must be able to see each listing stored in the database when searching without filters. Additionally, the user should be able to add, modify, and delete any listings they create in the database.
* Search for listing - Users must be able to search for listings using filters or inputted terms to find relevant listings. Any displayed data must be related to the search requirements to ensure integrity.

1. **Personnel management**

Brogan Murray - Frontend Developer / Backend Developer

Corban Larson - Backend Developer

Jake Cramer - Frontend Developer / Project Manager

Our primary source of communication is discord, as we have an established chat in place. We chose a Github repository to share code updates, project fulfillment, and communication between job roles. Additionally, we will host in-person meetings/workshops throughout the project to maintain a solid understanding of each other’s contributions, either by meeting at the library or at one’s dorm on campus after 5:50 PM on Mondays and/or Wednesdays.