University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 3

Date: Oct 25, 2024

Group Number: 89

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Charlotte Du	86020567	p4n7e	chardudev@outlook.com
Ivy Lin	37133345	y4e4v	ivy.lin128@gmail.com
Haider Khan	46046181	r1t2l	khanh3777@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

MS3:

2. Our project idea is a gaming community database that stores information about different video games and things related to their respective communities. The application will allow individuals to make queries on information for specific games or filter/discover games with similarities (genre, developer, platform, etc).

3. MS4: **Due: Nov,29**

GUI (front end): Ivy (using HTML, CSS, Javascript)

Backend: Charlotte, Haider (using SQL, Node.js)

Task breakdown:

GUI tasks:

- **Done asap:**
    - Setting up HTML, CSS for barebones front end
- **First week of november:**
    - Adding search bar for user input
    - Adding code for handling input to query database
    - Adding form validation for user input
    - Adding functionality for error handling for inputs
- **Second week of november:**
    - Adding filter options for entities/attributes
    - Adding code for handling filter selections to query database
    - Adding interactive button to search data
    - Adding button functionality to refresh data when clicked
- **Third week of november:**
    - Displaying data
    - Organizing + formatting data in tables to be easily interpretable for users
- **Final couple days:**
    - General formatting and changes to layout of front end
    - Testing GUI to ensure proper functionality with backend

Backend:

1. SQL Script to Create All Tables and Data

Haider and Charlotte (work together) - by Nov, 3

- Write the **SQL DDL statements** (CREATE TABLE) to set up the database schema.
- Include **constraints** to model the ER diagram (e.g., foreign keys, primary keys).

- Write **INSERT statements** to populate the tables with data.
- Include **DROP TABLE statements** to allow the script to be run multiple times.
- Implement ON UPDATE CASCADE in some way

2. Queries for CRUD Operations and Selection

Haider - by Nov 10 :

- **INSERT Operation (1 pt)**: Write the query for inserting data and reference where it's implemented.
- **DELETE Operation (1 pt)**: Write the query for deleting data and reference its implementation.
- **UPDATE Operation (1 pt)**: Write the query for updating data and reference its implementation.

Charlotte - by Nov 10

- **Selection Query (1 pt)**: Write a query to retrieve data using selection criteria and reference its implementation.
- **Projection Query (1 pt)**: Write a query to retrieve specific columns from a table and reference its implementation.

3. Complex Queries

Haider - by Nov 20

- **Join Query (1 pt)**: Write a query that performs a join operation and reference its implementation.
- **Aggregation with GROUP BY (1 pt)**: Write a query using GROUP BY and reference its implementation.
- **Aggregation with HAVING (1 pt)**: Write a query using GROUP BY with HAVING and reference its implementation.

Charlotte - by Nov 20

- **Nested Aggregation with GROUP BY (1 pt)**: Write a query that performs nested aggregation and reference its implementation.
- **Division Query (1 pt)**: Write a query that performs a division operation and reference its implementation.
- **Complex SQL Queries (2 pts)**: Include copies of SQL queries (2.1.7 - 2.1.10) with short descriptions for each.

4. Documentation

Everyone - by Nov 29th

- Write a **detailed description** of the project, including what the final project accomplished.
- Provide a description explaining if the final schema differs from the original schema. If there were changes, explain why.
- **Document the queries** listed in the rubric (INSERT, DELETE, UPDATE, Selection, Projection, Join, etc.).
- Make sure each query has a **clear reference** to where the query can be found in the codebase.
- Write a brief description for each complex query (up to 1-2 sentences) explaining what the query does.

MS5: Group demo: sign up for a demo slot

MS6: Individual and Peer Assessment